بسم الله الرحمن الرحیم

آموزش پروژه کتابخانه و فروشگاه مجازی کتاب بوسیله c#

نویسنده

خانم افسانه کتابی

فهرست

فاز اول طراحی.....۶
مقدمه.....۱۱
هدف از تهیه ی نرم افزار.....۱۲
ویژگی های خاص نرم افزار.....۱۳
راهنمای نصب نرم افزار.....۱۴
نگاه کلی به نرم افزار
فرم اولیه ی برنامه ی مدیر.....۱۵
فرم مدیریت کاربران.....۱۶
فرم مدیریت غرفه ها.....۱۷
فرم مدیریت انبار.....۱۸
فرم حسابداری.....۲۰
فرم گزارش گیری.....۲۱
فرم برنامه ی خریدار.....۲۲
فرم راهنما.....۲۳
فرم غرفه ی کتاب.....۲۴
فرم پوستر.....۲۶
فرم ارسال سفارش.....۲۷
فرم ثبت سفارش.....۲۷
فرم سبد خرید کالا.....۲۸
فرم ارایه ی خدمات به مشتری.....۲۹
فصل دوم
توابع مربوط به کلاس IRA
کد انتخاب رکورد.....۳۲
کد درج رکورد.....۳۴
تابع update.....۳۵
تابعdelet.....۳۷
تابعencode.....۳۹
تابعdecode.....۴۰
تابعshowdate.....۴۰

توابع مربوط به غرفه ی کتاب در برنامه ی مدیر

تابع bind..... ۴۳
تابع filter..... ۴۳
تابع اتصال به بانک اطلاعاتی..... ۴۴
تابع ثبت اطلاعات..... ۴۵
تابع price..... ۴۷
تابع check..... ۴۹
تابع clear..... ۵۰
تابع جستجو بر اسا س تعداد..... ۵۱
تابع جستجو بر اساس قیمت..... ۵۲
تابع جستجو بر اساس نام..... ۵۳
تابع جستجو بر اساس کد کتاب..... ۵۴
تابع جستجو بر اساس نویسنده ی کتاب..... ۵۵
تابع جستجو بر اساس مترجم کتاب..... ۵۶
تابع جستجو بر اساس ناشر..... ۵۷
تابع جستجو بر اساس تاریخ چاپ..... ۵۸
کد مربوط به مدیریت انبار..... ۵۹
کد فرم اصلی در برنامه ی مدیر..... ۶۰
کد ساعت..... ۶۴
تابع report..... ۶۶

کد مربوط به قسمت کاربران در برنامه ی مدیر

کد تعریف مدیر..... ۶۸
کد تعریف مشتری..... ۶۸
کد تعریف پرسنل..... ۶۸
کد راهنما..... ۶۹
تابع save..... ۷۰
تابع حذف کاربران..... ۷۱
تابع حذف با استفاده از link..... ۷۲
کد تخفیف..... ۷۳
کد انتخاب عکس..... ۷۴

کد مربوط به قسمت لوازم التحربر در برنامه ی مدیر
تابع bind.....۷۵

تابع filter.....۷۶
تابع clear.....۷۷
تابع save.....۷۸
اتصال به بانک اطلاعاتی.....۷۹
کد بررسی وجود رکورد.....۸۰
کد insert.....۸۱
تابع حذف با استفاده از link.....۸۳
تابع price.....۸۵
تابع جستجوبر اساس شماره.....۸۶
تابع جستجو بر اساس نام.....۸۷
توابع مربوط به پوستر در برنامه ی خریدار
تابع bind.....۸۸
تابع showpic.....۹۰
تابع نمایش تعداد عکس.....۹۲
تابع back.....۹۲
تابع viewimage.....۹۲
تابع nextimage.....۹۴
تابع previosimage.....۹۴
تابع update.....۹۶
تابع insert.....۹۶
تابع help.....۹۷
توابع مربوط به لوازم التحریر در برنامه ی خریدار
تابع bind.....۹۸
Link برو تو سبد.....۱۰۱
تابع viewimage.....۱۰۱
تابع مربوط به تغییر نوع کالا.....۱۰۲
تابع برو تو سبد.....۱۰۴
تابع update.....۱۰۵
تابع insert.....۱۰۶

تابع مربوط به وارد کردن تعداد.....۱۰۷
توابع مربوط به قسمت **Sendbasket**
کد کسر تعداد فروخته شده از تعداد موجود.....۱۰۸
کد ارسال سفارش برای خریداران عادی.....۱۰۹
کد ارسال سفارش برای خریداران مشترک.....۱۰۹
تابع برو تو سبد.....۱۱۰
کد مربوط به برنامه ی اصلی در فرم خریدار
کد مربوط به فراخوانی کلاس **ira**.....۱۱۳
کد مربوط به **login** سیستم.....۱۱۴
کد خروج از سیستم.....۱۱۵
کد زمان.....۱۱۶
کد نمایش سبد.....۱۱۶
کد مربوط به برنامه **custom_service**.....۱۱۸
نتیجه گیری.....۱۲۰
پیوست.....۱۲۱
منابع و ماخذ.....۱۲۲

فاز اول طراحی ERD :

۱- مشخص کردن موجودیت ها :

۱- مدیر

۲- کالا

۳- کارکنان

۲- مشخص کردن خصوصیات هر موجودیت :

۱- مدیر :دارای کلیه ی مجوز های دسترسی به همه ی غرفه ها

۲- کالا :کد گروه،کد کالا ،نام کالا ،تعداد کالا ،قیمت کالا و توضیحات اضافی در مورد کالا.

۳- کارکنان دارای مجوز های دسترسی که مدیر برای آنها تعیین می کند.

۳- تعیین خصوصیات مشتق شده :

شماره ی ارسالی برای سفارش کالا که به خریدار داده می شود یک صفت مشتق شده می باشد.

در این عبارت با گرفتن تاریخ فعلی در قالب یک دستور سال و ماه تاریخ جاری اخذ شده و به عنوان جهار عدد اول کد سفارش در نظر گرفته می شود.

سپس یک عدد که معرف شماره ی سبد خرید می باشد و سپس عدد دیگر که معرف تعداد خرید کنندگان از سبد فعلی می باشند.

مثلا اگر شماره ی سفارش یک مشتری ۸٦۰۷۱۵ باشد به معنی این است که این مشتری در سال ۸٦ و ماه هفتم ،پانزدهمین مشتری میباشد که از سبد اول خرید کرده است.

٤- تعیین وابستگی های وجودی :

در این سیستم هر سه موجودیت مدیر ، کالا و کارکنان موجودیت های مستقلی هستند و هیچ یک به دیگری وابسته نیست . در نتیجه وابستگی وجودی نداریم .

۵- مشخص کردن چندی ارتباط بین موجودیت ها :

ارتباط بین کالا و مدیر ، عضویت مدیر در فروشگاه ، یک ارتباط N:۱ است .

ارتباط بین کالا و کارکنان ، کارکردن کارکنان در فروشگاه ، یک ارتباط N:۱ است .

ارتباط بین فروشگاه و کالا ، موجود بودن کالا در فروشگاه و دسته بندی آن ها براساس نوع کالا ها ، یک ارتباط N:۱ است .

٦- تعیین Primary Key برای هر موجودیت :

موجودیت	Primary Key
مدیر	کلمه ی عبور
کالا	کد کالا
کارکنان	کلمه عبور

۱) جهت تبدیل رابطه هایی به صورت زیر (که تمامی رابطه های مشخص شده در این سیستم از این نوع هستند) ، Primary Key موجودیت با چندی ۱ را در جدول موجودیت با چندی N قرار می دهیم .

تعـــــــداد ۲

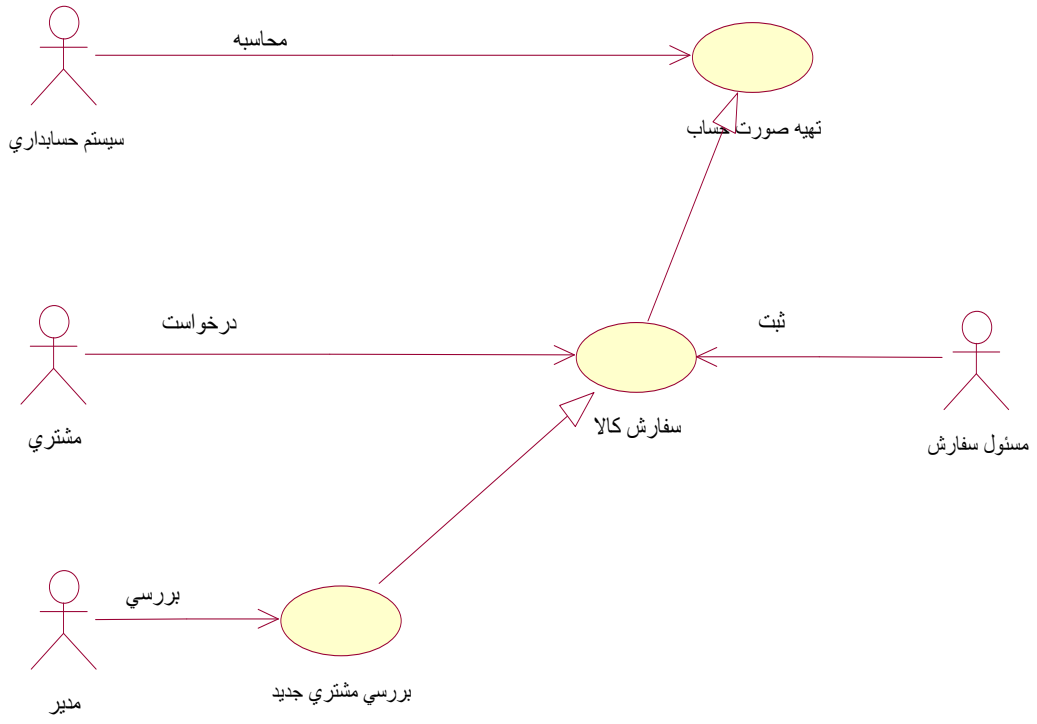
موجودیت

وضـــــعیت مستقل

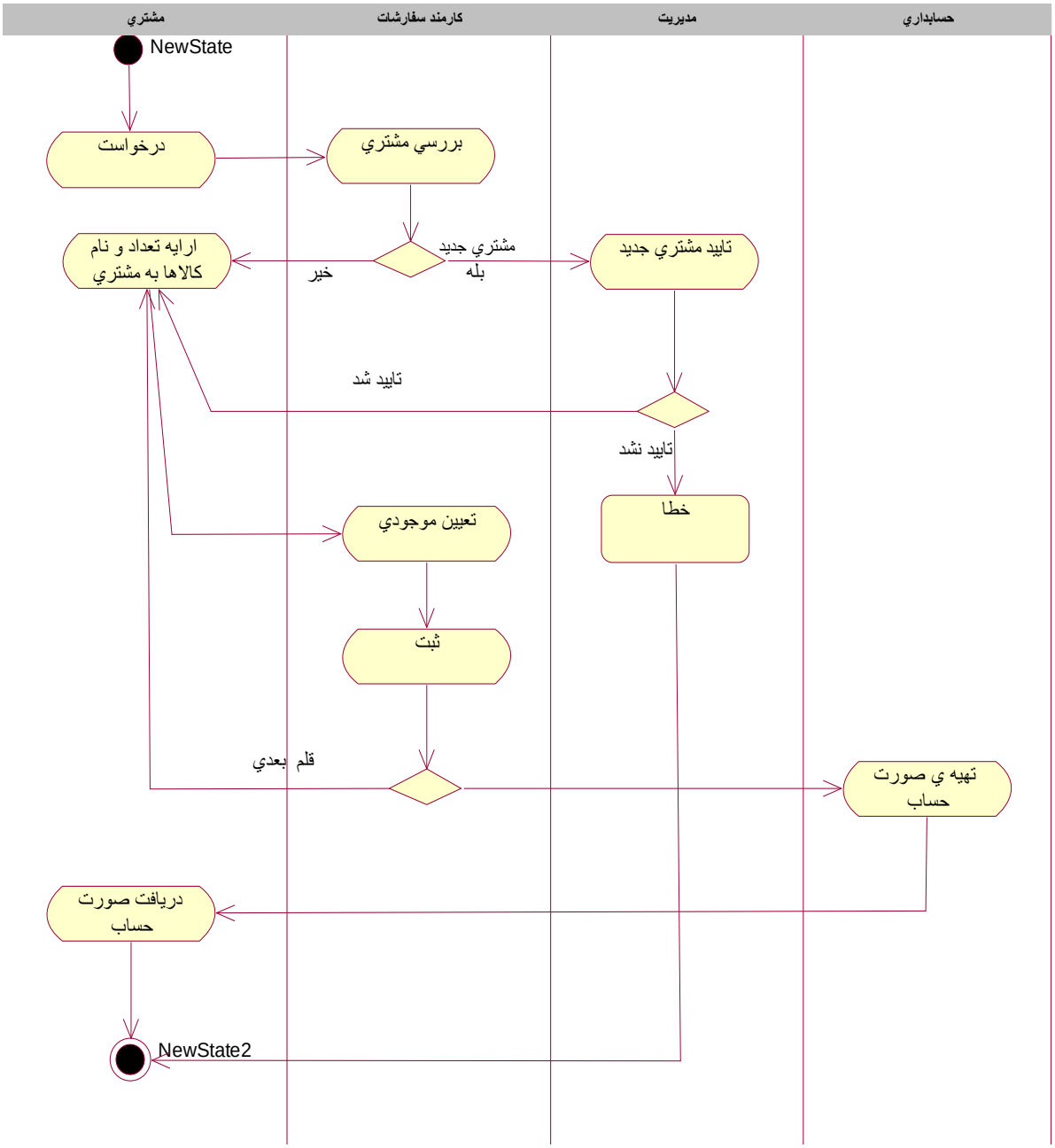
موجودیت ها

چندی رابطه ۱:N

به عنوان مثال : در رابطه کالا و اعضا ، کد کالا را در جدول موجودیت کتاب می گذاریم .



"Use case diagram"



"Activity Diagram"

فصل اول

مقدمه

شاخه ای از علم کامپیوتر برنامه نویسی است که هدف از آن تهیه نرم افزار است.یکی از اهداف مهم برنامه نویسی تولید نرم افزار های کاربردی است.نرم افزارهـای کـاربردی جهـت مکـانیزه نمـودن سیسـتمهای عملیاتی مختلف طراحی می شوند.
مکانیزه شدن سیستم های عملیاتی ادارات، نهادها و سایر اماکن اداری، صـنعتی،تجاری و ... دارای مزیـت هایی است که از جمله آنها را حذف کاغذ در انجام کارها، سرعت و دقت بـالای اجـرای عملیات،امنیـت اطلاعات و سادگی دسترسی به اطلاعات را می توان نام برد.
اینجانب نیز به عنوان دانشجوی رشته کامپیوتر امید است که توانسته باشم با طراحی این نرم افزار گامی را در این زمینه برداشته باشم.

هدف از تهیه نرم افزار

این نرم افزار جهت استفاده لوازم التحریرها و کتابفروشی ها طراحی شده است کـه دارای قابلیـت هـای گزارش گیری اطلاعات، امنیت دسترسی به اطلاعات، حسابداری و ... می باشد. این نـرم افـزار همچنـین دارای ابزارهای جانبی تنظیم ساعت وتاریخ و پشتیبان گیری اطلاعات می باشد.
این نرم افزار طوری تهیه شده که تمامی نیاز های یک فروشگاه بزرگ کتاب و لوازم التحریر امـروزی را فراهم می آورد. در این نرم افزار سعی بر ان شده است که ٤ فروشگاه مجزای کتاب ،لوازم التحریر،رسانه های آموزشی و کارت پستار ونقشه و ... را در یک فروشگاه بزرگ جمع آوری کنیم.

ویژگی ها خاص نرم افزار

این نرم افزار از ۳ برنامه ی کاملا مستقل از هم تشکیل شده که هر یک وظیفه ی خاصی را بر عهده دارند.در برنامه ی مربوط به مدیر بعد از اینکه مدیر با کلمه ی رمز شناسایی شد قادر به انجام کلیه ی کارها مربوط به کاربران ،غرفه ها ، انبار و حسابداری می باشد.
در برنامه خریدار امکان خرید کالا برای کسانی که برای اولین بار به فروشگاه ما وارد شده اند و یا کسانی که مشترک فروشگاه ما می باشند وجود دارد.بدیهی است که گروه دوم از تخفیف های خاص خود بهره مند می شوند.
و در برنامه ی سوم سرویسی که به خریدار داده شده است برای او و مدیر فروشگاه قابل شهود می باشد.
از ویژگی های خاص نرم افزار می توان ابزار جانبی سیستم پشتیبان گیری، سیستم امنیت ورود به برنامه با ویژگی کد کردن کلمه عبور،سیستم کنترل تعداد کالاهای انبار، دادن هشدار مربوط به کمبود تعداد کالا ،دادن سبد خرید خاص به هر مشتری میباشد.

راهنمای نصب و استفاده از نرم افزار

۱- سیستم عامل و سخت افزار مورد نیاز

- **کلیه نسخه های سیتم عامل ویندوز XP**
- **پردازنده پنتیوم ۳ به بالا**
- **حداقل حافظه مورد نیاز MB۱۲۸**

۲- طریقه ی نصب نرم افزار

برای نصب نرم افزار فایل "adminestator" را در پوشه ی "bookshopl" موجـود بـر روی سـی دی اجرا کنید تا قسمت مدیریت را به اجرا در آورید و برای اجرای برنامه ی خریـدار فایـل "customer" واقع در پوشه ی "bookshop" و اجرای سرویس خدمات فایل"customer_service" واقع در پوشه ی "bookshop" را نصب کنید.

نگاه کلی به نرم افزار:

این نرم افزار از سه برنامه ی مستقل تشکیل شده که شرح برنامه ها به صورت زیر می باشد:

برنامه ی مدیر:

فرم اولیه ی تشکیل دهنده ی این برنامه یک فرم ساده می باشد .در این فرم پنج button وجود دارد که در حالت عادی غیر از دکمه ی خروج بقیه ی دکمه ها غیر فعال می باشد.
در این فرم برای لحاظ امنیت بیشتر فقط کسانی می توانند وارد سیستم شوند که دارای نام کاربری و شماره ی رمز باشند.اگر کاربر به عنوان مدیر وارد سیستم شود و نام کاربری و رمز او قابل قبول باشد ضمن خوش آمد گویی تمامی Button ها برای او فعال می شود.این دکمه ها عبارتند از غرفه ها، کاربران،انباروحسابداری .
اگر کاربر به عنوان خریدار و یا پرسنل وارد برنامه شود به او پیغام داده خواهد شد که مجوز دسترسی ندارد.

فرم اولیه ی برنامه ی مدیر:

کاربر بعد از نوشتن نام کاربری و رمز عبور با کلیک بر روی دکمه ی ورود به سیستم می تواند وارد برنامه شود.در اینجا فرض می کنیم که یک کاربر مدیر به سیستم login کرده است.

با اختیاراتی که برای مدیر وجود دارد او می تواند مدیریت تمامی قسمت ها را بر عهده بگیرد.یکی از این قسمت ها مدیریت کاربران می باشد.

در این قسمت همانطور که مشاهده می شود مدیر می تواند برای فروشگاه مشتری ، پرسنل و یا مدیر تعریف کند.در قسمت نوع کاربر کاربری که قرار است به فروشگاه اضافه شود تعیین می شود .

در این قسمت اگر یک کاربر مشتری تعریف شود وی مجوز دسترسی نخواهد داشت.اگر مشتری در حال تعریف باشد قسمت مربوط به تخفیف فعال می شود و مدیر می تواند به هر مشتری تخفیف لازم را بدهد.

اگر یک پرسنل برای فروشگاه تعریف شود قسمت مربوط به مجوز دسترسی پرسنل فعال می شود و مدیر با توجه به نیاز می تواند به وی مجوز دسترسی های لازم را بدهد.

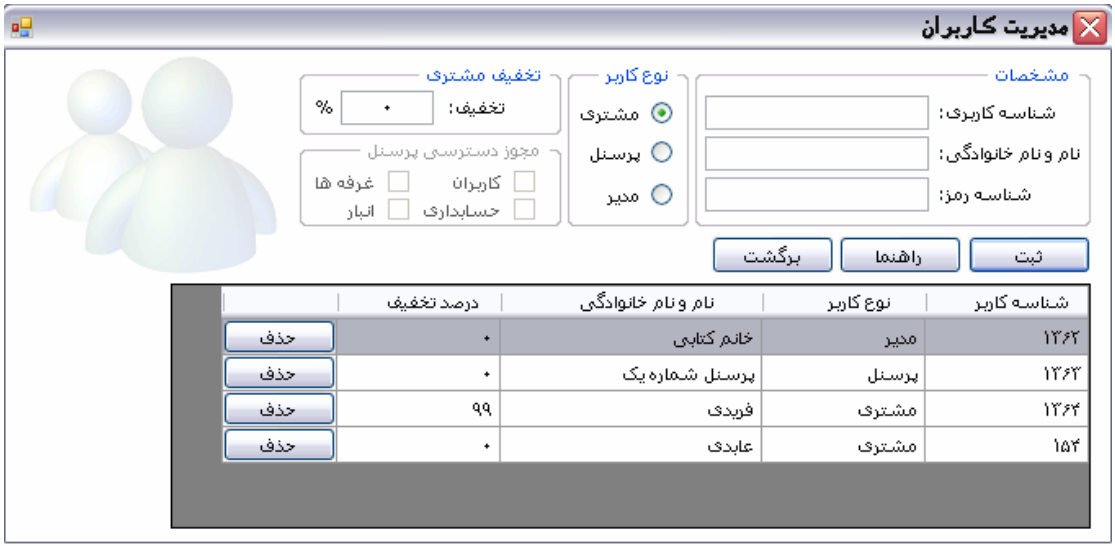
همچنین مدیر می تواند برای فروشگاه مدیر یا مدیران دیگری را نیز تعریف نماید.

وقتی که قسمت تعریف کاربر به اتمام رسید با زدن دکمه ی ثبت اطلاعات وارد شده ثبت می گردد.

هم چنین برای راهنمایی و آگاهی از چند و چون کار می تواند از راهنمای مربوط به فرم استفاده کرد.

با زدن دکمه ی برگشت به فرم اولیه ی برنامه باز می گردیم.

فرم مربوط به قسمت مدیریت کاربران:



فرم مربوط به مدیریت غرفه ها:

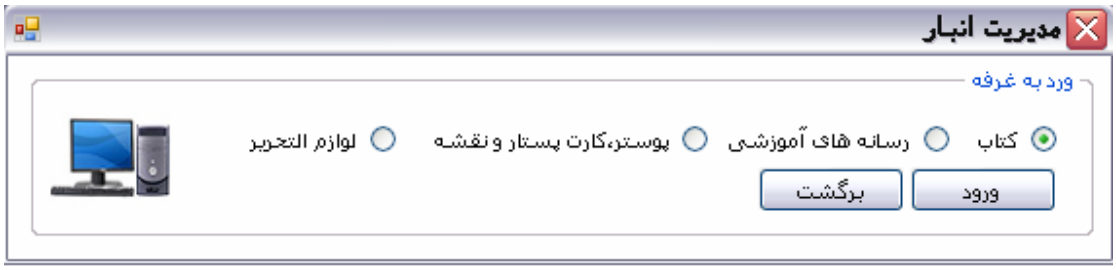
این فروشگاه لوازم التحریر و کتاب از ٤ غرفه ی کتاب ، رسانه های آموزشی ،پوستر و نقشه و لوازم التحریر تشکیل شده است.

طریقه ی شناسایی کالا در این فروشگاه به این صورت است که برای هر غرفه یک شماره به نام شماره ی گروه و برای تک تک کالاهای موجود در غرفه یک شماره به نام شماره ی کالا در نظر گرفته شده است.برای شناسایی یک کالا ابتدا از طریق شماره ی گروه و سپس از طریق شماره ی کالا پی گیری می شود.در اینجا با انتخاب هر غرفه ایی می توان کالا ی مورد نظر را به ثبت رساند.مثلا با انتخاب غرفه ی کتاب می توان یک گروه به این غرفه به نام گروه کتابهای تخصصی کامپیوتر افزود.سپس با زدن دکمه ی ثبت اطلاعات جدید در پایگاه داده به ثبت خواهد رسید.می توان از دکمه ی راهنما برای کمک گرفتن انجام عملیات استفاده کرد و با زدن دکمه ی برگشت به برنامه ی اول بازگشت.

فرم مدیریت غرفه ها:

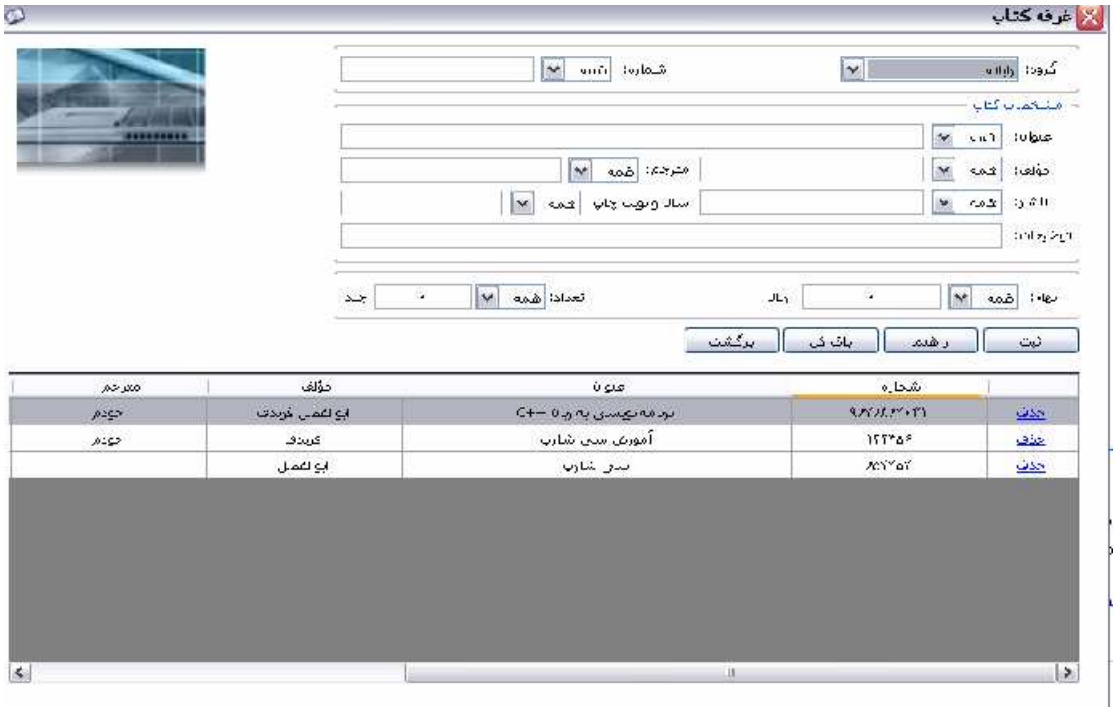
فرم مربوط به مدیریت انبار:

با کلیک کردن بر روی دکمه ی مدیریت انبار فرمی مشابه فرم زیر باز میشود:



که نمایانگر این است که به انبار کدام یک از غرفه ها می خواهید دسترسی پیدا کنید؟ به عنوان مثال برای رفتن به انبار مربوط به غرفه ی کتاب روی آن کلیک کرده و دکمه ی ورود را می زنیم.

فرمی شبیه فرم زیر باز می شود:

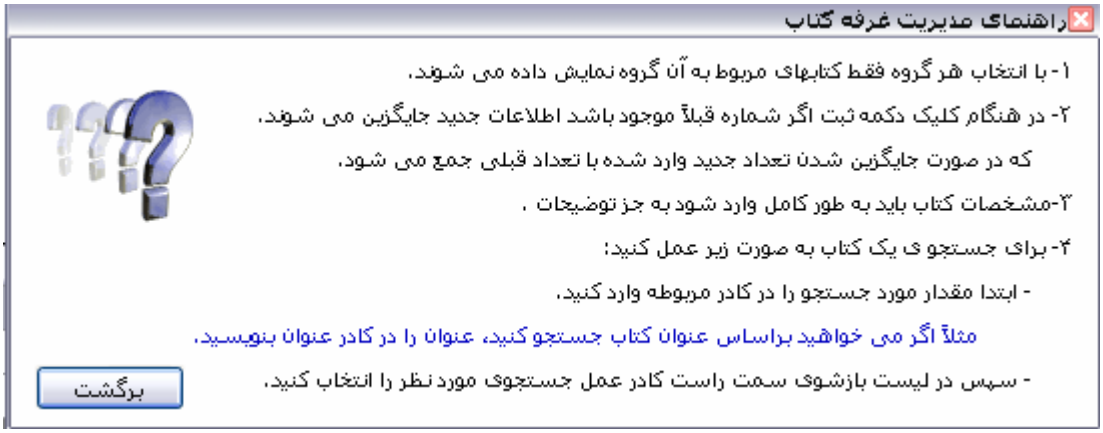


در این فرم با وارد کردن اطلاعات کتاب که شامل نام کتاب، نویسنده،مترجم ،سال چاپ ، بها و قیمت آن می باشد می توان کتابی را در فروشگاه به ثبت رسانید.نکته ی قابل توجه در این فرم این است که در قسمت تعداد و قیمت نمی توان از حروف و یا عملیات کپی استفاده کرد.

با این کار درصد خطای کاربر را به حداقل می رسانیم.

هم چنین در این قسمت می توان به دنبال کالای مشخصی نیز گشت.یعنی امکان جستجو را برای ما فراهم می آورد.به این ترتیب که در لیست بازشوی سمت راست این گزینه ها ۳ گزینه به نام همه ، شبیه و برابر به چشم می خورد . اگر دنبال کالایی هستید که دقیقا برابر با نامی می باشد که در اختیار دارید ار گزینه ی برابر و اگر می خواهید شبیه به چیزی باشد که وارد شده است از گزینه ی شبیه استفاده کنید.

این قسمت نیز مانند قسمت های دیگر با زدن دکمه ی ثبت اطلاعات را ثبت کرده و با زدن دکمه ی راهنما فرمی مانند فرم زیر برای راهنمایی کاربر باز می شود:



فرم مربوط به قسمت حسابداری:

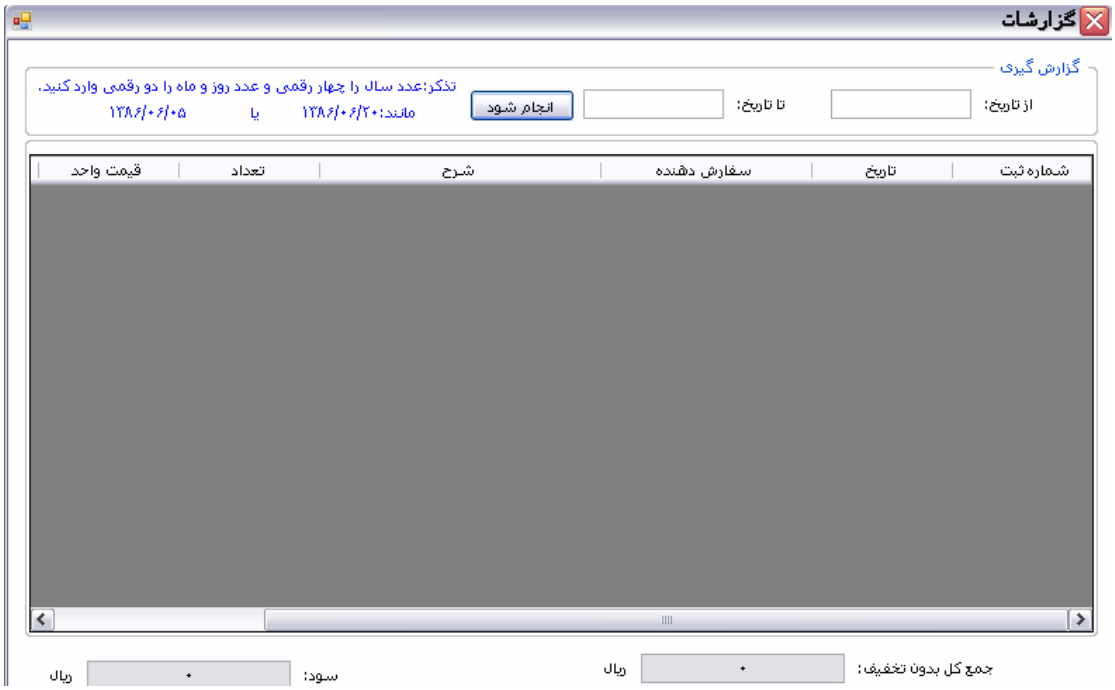
با انتخاب قسمت حسابداری فرمی مشابه فرم زیر باز می شود:

همانطور که مشاهده می شود این قسمت به انجام امور حسابداری چون تعیین نرخ سود و انجام عمل گزارش گیری می پردازد.

با انتخاب قسمت نرخ سود فرمی مشابه فرم زیر باز می شود:

که در این قسمت میزان نرخ سود هر غرفه مشخص می شود.

با زدن گزینه ی گزارش گیری فرمی شبیه فرم زیر باز می شود:



که در این قسمت کلیه ی امور گزارشگیری انجام می شود.برای ا نجام عمل گزارشگیری کا فی است کـه تاریخ مورد درخواست را به طریقه ی گفته شده وارد کنید و گزارش انجام کارها را مشاهده کنید.
این گزارش ها شامل نمایش تخفیف داده شده ، فروش بدون در نظر گرفتن تخفیف ، بـا در نظـر گـرفتن تخفیف و هم چنین نرخ سود می باشد.

برنامه ی خریدار:

فرم اصلی این برنامه یک فرم ساده به صورت فرم زیر می باشد:

در این فرم ضمن خوش آمد گویی به مشتری مقررات موجود در فروشگاه را به وی یاد آور شده است.
ویژگی که در این فرم به چشم می خورد این است که مشتری نمی تواند این فرم را ببندد تا مشتری دیگر برای ورود به سیستم با مشگل مواجه نشود.
با کلیک بر روی link ورود به سیستم فرمی به شکل زیر ظاهر می شود:

با وارد شدن به این قسمت اگر مشتری با روال کار فروشگاه آشنایی نداشته باشد می تواند باا رفتن به قسمت راهنمایی خرید با روال کار سیستم آشنا شود.این فرم به شکل زیر می باشد:

راهنمای خرید

مشتری گرامی:

برای خرید مراحل زیر را به ترتیب انجام دهید:

۱- ابتدا با کلیک بر روی دکمه ورود به سیستم در صفحه اولیه وارد سیستم شوید.

- پس از ورود به سیستم یک سبد خرید برای شما ساخته می شود.

۲- با ورود به هر غرفه کالای مورد نیاز خود را به تعداد دلخواه وارد سبد کنید.

- وارد نمودن کالا در سبد با کلیک دکمه "برو تو سبد" در هر غرفه صورت می گیرد.

۳- پس از آماده نمودن سبد با کلیک دکمه مشاهده سبد می توانید سبد خرید را ببینید.

۴- برای آماده شدن سفارش باید سفارش را در قسمت ارسال سفارش ارسال کنید. برای این کار:

- اگر مشترک هستید شناسه اشتراک و شناسه رمز خود را وارد کنید.

- اگر مشترک نیستید فقط نام و نام خانوادگی خود را در کادر مربوطه وارد کنید.

- پس از ارسال منتظر دریافت شماره ثبت سفارش بمانید.

- پس از دریافت شماره ثبت سفارش دکمه "ادامه" را کلیک کنید.

- برای تحویل گرفتن کالا شماره ثبت سفارش را به باجه تحویل ارائه داده و اجناس خود را تحویل بگیرید.

برگشت

این فرم دارای گزینه هایی می باشد که با آنها آشنا می شویم.در ابتدا غرفه های موجود در فروشگاه نمایش داده می شود و کاربر با توجه به نیاز خود برای خرید به یکی از این غرفه ها مراجعه می کند.به عنوان مثال برای خرید کردن کتاب این غرفه را انتخاب می کنیم و به قسمت خرید می رویم.فرم این قسمت به این صورت می باشد:

به غرفه کتاب خوش آمدید — شماره سبد شما:1

گروه: رایانه — مشاهده پیغام درج در سبد

تعداد مورد نیاز: ۰

	شماره	عنوان	مؤلف	مترجم	ناشر
برو تو سبد	۹۶۴۶۸۶۴۰۳۱	برنامه نویسی به زبان C++	ابوالفضل فریدی	خودم	علوم رایانه
برو تو سبد	۱۲۳۴۵۶	آموزش سی شارپ	فریدی	خودم	همه
برو تو سبد	۶۵۴۴۵۴	سی شارپ	ابوالفضل		

مشاهده سبد خرید — راهنمای خرید — برگشت

در این قسمت با انتخاب کردن نوع کتاب ها کتاب هایی که از آن نوع وجود دارند به نمایش گذاشته می شوند.مثلا در فرم بالا کتاب هایی از نوع رایانه انتخاب شده اند.
همانطور که مشاهده می شود به همراه شماره کتاب نام و قیمت و مولف آن نیز نمایش داده می شود.کاربر برای خرید کالا باید تعداد مورد نیاز خود را وارد کند.اگر تعداد وارد نشود با یک پیغام متذکر می شود که تعداد وارد نشده و یابد تعداد زا وارد کنیدوهم جنین برای نوشتن تعداد نمی توان از حروف و یا عملیات کپی استفاده کرد.بعد از انتخاب تعداد با انتخاب گزینه ی برو تو سبد تعداد مورد نیاز وارد سبد خرید می شود مبنی بر اینکه تعداد چند کتاب در سبد خرید قرار داده شده است به صورت شکل زیر: بدیهی است که هر کاربری که وارد سیستم می شود یک سبد خرید با یک شماره ی خاص به وی اختصاص داده می شود .این شماره به هیچ وجه بین دو مشتری تکراری نمی شود.با خروج مشتری از فروشگاه این سبد از وی گرفته شده و به یک مشتری دیگر داده می شود.

برای رفتن به غرفه ی پوستر و نقشه این غرفه را انتخاب کرده و وارد قسمت خرید آن می شویم.در این فرم باید مشخص کنیم که چه نوع کالایی را مدنظر داریم.کالاها در این قسمت به ۳ دسته ی پوستر ، کارت پستار و نقشه تقسیم می شود.هر کدام ار این گروه ها باز به گروه های دیگری تقسیم بندی می شوند.مثلا پوستر ها شامل پوستر های طبیعت ، تاریخی و... می باشند.در این فرم نام کالا ،قیمت و توضیحات و... برای هر کالا به چشم می خورد.مانند غرفه های دیگر برای خرید کالا باید تعداد را مشخص کرد و سپس linke برو تو سبد را فعال کرد.
در این فرم با انتخاب گزینه ی تصویر بعدی میتوان تصویر بعد از تصویر فعلی و با انتخاب گزینه ی تصویر قبلی تصویر قبل از تصویر فعلی را مشاهده کرد.

هم چنین می توان در هر فرمی که هستید سبد خرید خود را مشاهده کنید که ا کنون چه خرید هایی را انجام داده اید.

با انتخاب قسمت راهنما در هر فرم می توان از راهنمای خود سیستم استفاده کرد.

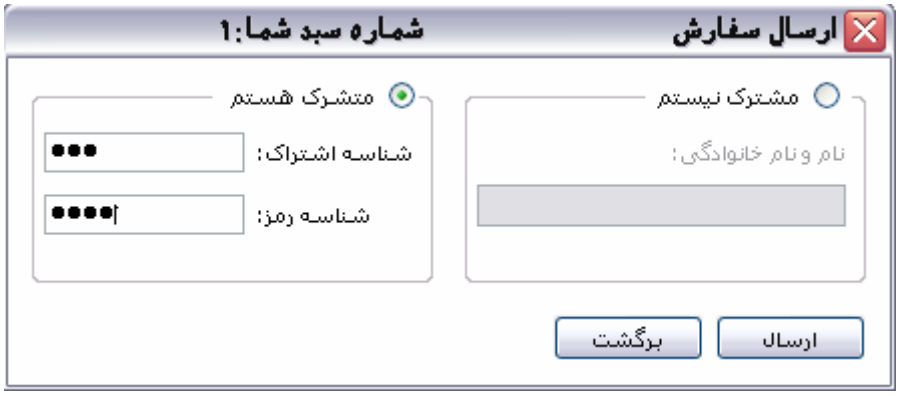
نمونه ایی از فرم پوستر:

بعد از اینکه خرید را انجام داده اید و دیگر خریدی ندارید وقت آن است که با پرداخت هزینه ی خرید های خود را تحویل بگیرید.برای اینکار از گزینه ی ارسال سفارش استفاده کنید.

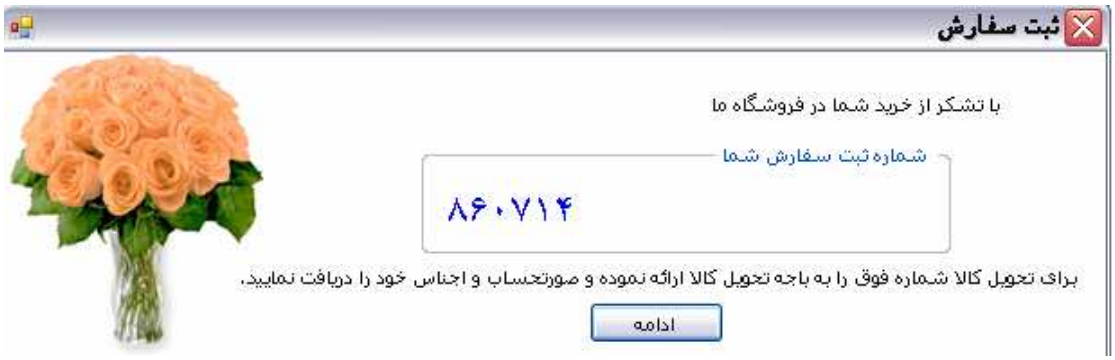
در فرمی که ظاهر می شود باید مشخص کنید که آیا مشترک این فروشگاه هستید و یا برای اولین بار از این فروشگاه خرید می کنیدوبدیهی است که اگر مشتری دایمی فروشگاه باشید از تخفیف نیز بهره مند خواهید شد.

با انتخاب گزینه ی مشترک هستم باید شماره و رمز خود را وارد کنید.

فرم این فسمت به صورت زیر می باشد:

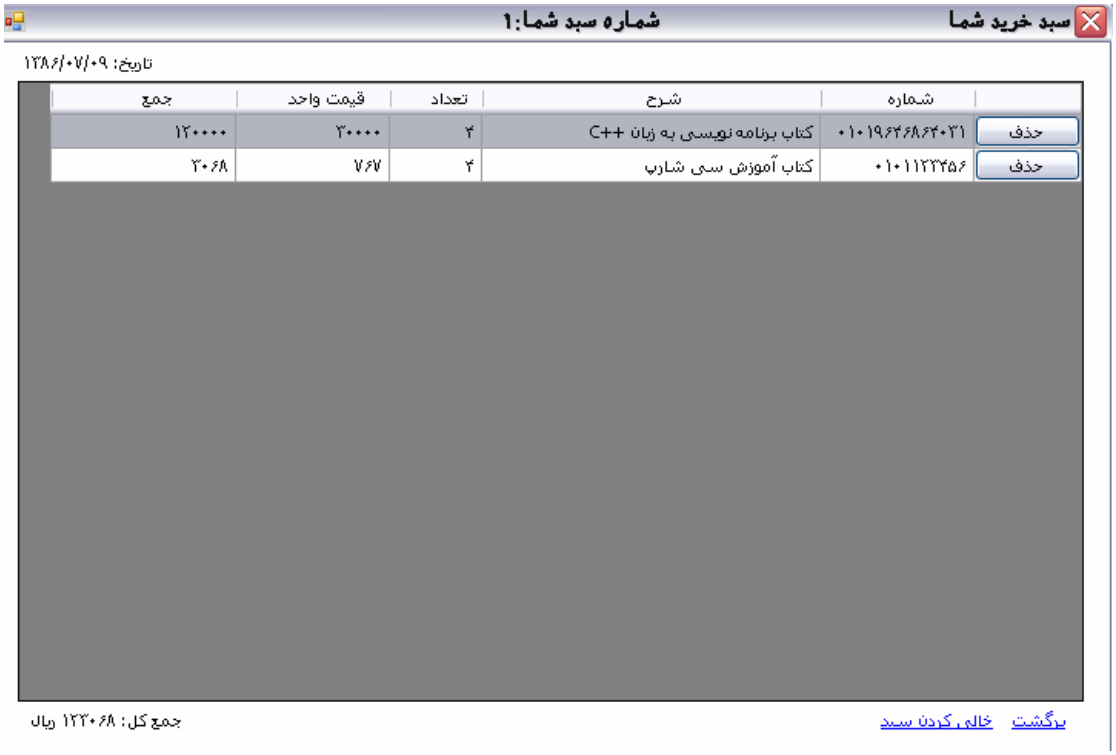


بعد از اینکه نام و نام خانوادگی و یا شناسه ی اشتراک خود را وارد کردید فرمی به صورت روبرو باز می شود:



همانطور که در فرم نیز گفته شده باید برای تحویل کالا شماره ی فوق را به باجه تحویل کالا ارایه داده و صورتحساب و اجناس خود را دریافت نمایید.
آخرین گزینه ایی که در این قسمت وجود دارد فرم مشاهده ی سبد می باشد که با زدن این دکمه سبد خرید شما نمایش داده می شود.

نمونه ایی از سبد خرید کالا:



برنامه ی دیگر خدمات ارایه شده به مشتری را نمایش می دهد.در این خدمات به مشتری گفته شده که چه چیز هایی با چه قیمت و چه تخفیفی انتخاب کرده است و در نهایت هزینه ی کل ارایه ی خـدمات را بـه وی را نمایش می دهد.

فرم مربوط به ارِایه ی خدمات به مشتری:

لیست سفارشات

شماره ثبت سفارش: ۸۶۰۷۱۵ نمایش

تاریخ: ۱۳۸۶/۰۷/۰۹

خریدار: کتابی

تخفیف: ۰ %

ردیف	شرح	تعداد	قیمت واحد	جمع
۱	کتاب برنامه نویسی به زبان C++	۴	۳۰۰۰۰	۱۲۰۰۰۰
۲	کتاب آموزش سی شارپ	۴	۷۶۷	۳۰۶۸
۳	کتاب سی شارپ	۴	۲۰۰۰۰	۸۰۰۰۰

جمع کل: ۲۰۳۰۶۸ ریال

فصل دوم

توابع و دستورات کلی نرم افزار

تمامی این ۳ برنامه از یک پایگاه داده استفاده می کند و لذا پایگاه داده را برای یک بار و برای کلیه ی برنامه ها تعریف کرده و سپس در هنگام نیاز به آن مراجعه می کنیم.به همین دلیل ابتدا نحوه ی اتصال به پایگاه داده را شرح می دهیم.برای ارتباط با پایگاه داده در یک محیط شی گرا مانند #c از using مربوط به آن یعنی sqlclient استفاده می شود.
کلاس IRA مهم ترین کلاس در برنامه فروشگاه می باشد که نحوه ی برقراری با پایگاه داده را شرح می دهد.

```
class IRA
{
SqlConnection Connection = new SqlConnection("Data Source=localhost;Initial Catalog=Shop;Integrated Security=True");
SqlCommand Command = new SqlCommand();
SqlDataAdapter Adapter = new SqlDataAdapter();
DataSet DS = new DataSet();
```

در ابتدا باید ارتباطی بین برنامه و بانک اطلاعاتی به وجود بیاوریم که این ارتباط از طریق "sqlconnection" ایجاد می شود پس با دستور اول از تابع فوق یک شی به نام "connection" برای ایجاد ارتباط با بانک اطلاعاتی درست می شود.در هنگام ایجاد این ارتباط "data source" را هم مشخص می کنیم."data source" آدرس و نام پایگاه داده را مشخص می کند که در این برنامه "shop" نام گذاری شده است.در دستور بعدی برای اجرای دستورات sql از "sqlcommand" استفاده کردیم.برای استفاده از" sql command" نیز یک شی به نام "command"ساخته شده است.
توسط "SqlDataAdaptor" میتوان درخواست یا تقاضای خود را از موتور بانک اطلاعاتی کرد که برای استفاده از این دستور نیز مانند دستورات فوق یک شی از آن ایجاد می کنیم.
یکی از جالب ترین اجزا"ado.net" قسمت"dataset" آن می باشد. "Data set" برای نگهداری اطلاعات به کار می رود یکی از محاسن "Data set"این است که توسط آن می توانیم چندین نوع اطلاعات رادر قالب جدول های مجازی ذخیره کنیم.
برای نگهداری اطلاعات در داخل "Dataset" از متد fillاستفاده می شود.
اکنون وقت آن رسیده که یک دستور sql را از ورودی بگیریم و آن را با استفاده از پایگاه داده به اجرا در آوریم که این کار با استفاده از دستورات زیر امکان پذیر می باشد.

```
//-- Execute SQL Command
private void ExecuteSQLCommand(string _CmdText)
```

```
{
Command.CommandType = System.Data.CommandType.Text;
Command.CommandText = _CmdText;
Command.Connection = Connection;
Connection.Open();
Command.ExecuteNonQuery();
Connection.Close();
}
```

_cmdtext ورودی دستور sql می باشد که به صورت رشته وارد شده است.

با دستور Command.CommandText = _CmdText; ورودی دستور sql را به شی "command" می دهیم که ورودی را به اجرا در آورد.

شی "command" که آماده ی خواندن از پایگاه داده می شود باید بداند که از چه طریقی می خواهد به پایگاه داده وصل شود یا به عبارتی ساده واسط این دستور را برای استفاده از پایگاه داه را بشناسد که برای این منظور از دستور Command.Connection = Connection; استفاده شده است.

دستور بعدی connection را باز کرده و بعد از اجرای دستور آن را می بندیم.

فرم مربوط به انتخاب رکوردها:

```
//-- Select record from database and puts in dataset
public void Select_Record(string _TableName, string _Fields, out DataSet _DS, string _ConditionalFileds,
string _ConditionalCompareOperator, string _ConditionalLogicOperator, params string[]
_ConditionalValues)
{
string cmdtext = "";
if (_ConditionalFileds != string.Empty)
{
cmdtext = "SELECT " + _Fields + " FROM " + _TableName + " WHERE ";
string prm = "";
string[] Fields = new string[100];
int i = 0;
foreach (char c in _ConditionalFileds)
{
if (c != ',')
Fields[i] += c;
else
i++;
}
i = 0;
Command.Parameters.Clear();
foreach (string Value in _ConditionalValues)
{
prm = "@Val_" + i.ToString();
Command.Parameters.Add(prm, SqlDbType.NVarChar);
Command.Parameters[prm].Value = Value.Trim();
```

```
cmdtext += Fields[i] + " " + _ConditionalCompareOperator + " " + prm + " " + _ConditionalLogicOperator
+ " ";
i++;
}
cmdtext = cmdtext.Substring(0, cmdtext.Length - (_ConditionalLogicOperator.Length + 2));
}
else
cmdtext = "SELECT " + _Fields + " FROM " + _TableName  ;
DS.Clear();
Command.CommandText = cmdtext;
Command.Connection = Connection;
Connection.Open();
Adapter.SelectCommand = Command;
Adapter.Fill(DS);
Connection.Close();
_DS = DS;
}
```

تابع فوق تعدادی رکورد را در یک جدول انتخاب می کندو رکورد های انتخاب شده را در یک Data "set" قرار می دهد.انتخاب شدن رکوردها با شرطی که ما تعیین می کنیم انجام می شود.

پارامتر هایی که در این تابع استفاده شده به این شکل میباشند:

_tablename:نام جدولی که رکوردها در آن قرار داده شده اند را مشخص میکند و یک پارامتر ورودی می باشد.

_ fields:نام فیلد هایی از رکورد است و اگر * نوشته شود تمامی فیلد ها خوانده می شود و نوع آن ورودی می باشد.

_conditionalfield:شرط تعیین رکوردها را انتخاب می کند که می تواند نام دو یا چند فیلد باشداگر مقدار تهی به این پارامتر داده شود شرط در نظر گرفته نمی شود و تمامی رکوردهای جدول خوانده می شوند.

_ds:یک Data setاست که رکوردهای انتخاب شده در آن قرار می گیرند و نوع آن خروجی می باشد.

_conditional compare operator:عملگر مقایسه ی شرط است که می تواند هر یک از عملگرهای مقایسه ایی در sql باشد.مانند <= ،=>،<>،like و...

_conditional logic operator:نام یک عملگر منطقی مانند and وor است که فقط زمانی کاربرد دارد که فیلد شرط بیشتر از یک فیلد باشد ونوع آن ورودی است.

_conditional value:مقادیر را برای فیلد های شرطی تعیین می کند.

بررسی برای وجود شرط انتخاب توسط این دستور انجام می پذیرد.شرط انتخاب شرطی است که رکورد ها توسط آنان ا ز یکدیگر جدا می شوند.از کل رکورد هایی که در پایگاه داده داریم ما فقط به آنهایی نیازمندیم که یک یا چند شرط را دارا باشند لذا اول باید بررسی کنیم که اصلا روی خواندن رکوردها شرطی وجود دارد یا قرار است تمامی رکوردها خوانده شوند.که قسمت اول تابع به این شکل می باشد:

```
if (_ConditionalFileds != string.Empty)
```

```
{
cmdtext = "SELECT " + _Fields + " FROM " + _TableName + " WHERE ";
string prm = "";
string[] Fields = new string[100];
int i = 0;
foreach (char c in _ConditionalFileds)
{
if (c != ',')
Fields[i] += c;
else
i++;
}
i = 0;
Command.Parameters.Clear();
foreach (string Value in _ConditionalValues)
{
prm = "@Val_" + i.ToString();
Command.Parameters.Add(prm, SqlDbType.NVarChar);
Command.Parameters[prm].Value = Value.Trim();
cmdtext += Fields[i] + " " + _ConditionalCompareOperator + " " + prm + " " + _ConditionalLogicOperator
+ " ";
i++;
}
cmdtext = cmdtext.Substring(0, cmdtext.Length - (_ConditionalLogicOperator.Length + 2));
}
```

قسمت else این شرط حالتی را بررسی می کند که دستور select بدون where باشد ،یعنی هیچ گونه شرطی اعمال نشده . منظور تمامی رکوردها است.

Ds.clear:پاک کردن رکورد های قبلی.

Command.commandtext=cmdtext:در اینجا دستوری را که در قسمت بالا در رشته ی "cmdtext" قرار دادیم را به "command" می دهیم که آن را به اجرا در آورد و چون برای استفاده از "command" هم یک ارتباطی با بانک اطلاعاتی باید به وجود آید از command.connection=connection استفاده کردیم.

Connection.open():برای باز کردن بانک اطلاعاتی.

Adapterselectcommand=command: برای خواندن رکوردها از این دستور استفاده می شود.

Adapter.fill(ds):با این دستور رکوردها را داخل dataset قرار می دهیم.

Connection.close();:برای بستن بانک اطلاعاتی استفاده می شود.

ds=ds-:برگرداندن dataset به برنامه فراخوان.

تابع بعدی برای درج یک رکورد در جدول می باشد:

```
//-- Insert Record In database
public void Insert_Record(string _TableName,string _Fileds, params  string[] _Values)
{
string cmdtext = "INSERT INTO " + _TableName +"(" + _Fileds + ") VALUES(";
```

```
string prm = "";
Command.Parameters.Clear();
int i = 0;
foreach (string Value in _Values)
{
prm = "@Val_" + i.ToString();
Command.Parameters.Add(prm, SqlDbType.NVarChar);
if (Value != null)
Command.Parameters[prm].Value = Value.Trim();
else
Command.Parameters[prm].Value = "";
cmdtext += prm + ",";
i++;
}
cmdtext = cmdtext.Substring(0, cmdtext.Length - 1);
cmdtext += ")";
ExecuteSQLCommand(cmdtext);
}
```

پارامتر های تابع

_tablename:نام جدول را مشخص می کند.
_filds:اسامی فیلد ها را مشخص می کند.
_values:مقادیر فیلدها را مشخص می کند.

```
string cmdtext = "INSERT INTO " + _TableName +"(" + _Fileds + ") VALUES(";
```

به وسیله ی این خط برنامه یک دستور sql ایجاد کردیم که به پایگاه داده می گوید در جدولی به نام مشخص فیلد هایی با مشخصات مشخصی را درج کن.
در قسمت foreach برنامه مقایر را به صورت پارامتر برای فیلد ها قرار می دهیم که sqldbtype.nvarchar نوع داده پارامتر و prm نام پارامتر می باشد.
با دستور(executesqlcommand(cmdtext دستور sql اجرا می شود.

تابع دیگر تابع update می باشد:

```
Update Record From Database
public void Update_Record(string _TableName, string _Conditional, string _Fileds, params string[] _Values)
{
string[] Fields = new string[100];
int i=0;
foreach (char c in _Fileds)
{
if (c != ',')
Fields[i] += c;
else
i++;
}
string cmdtext = "UPDATE " + _TableName + " SET ";
string prm = "";
Command.Parameters.Clear();
i = 0;
foreach (string Value in _Values)
{
prm = "@Val_" + i.ToString();
Command.Parameters.Add(prm, SqlDbType.NVarChar);
if(Value != null)
Command.Parameters[prm].Value = Value.Trim();
else
Command.Parameters[prm].Value = "";
cmdtext += Fields[i]+ "=" + prm + ", " ;
i++;
}
cmdtext = cmdtext.Substring(0, cmdtext.Length - 2);
cmdtext += " WHERE " + _Conditional;
ExecuteSQLCommand(cmdtext);
}
```

تابع update یک یا تعدادی از رکوردهای یک جدول را به روز رسانی می کند.

پارامتر های این تابع به شرح زیر است:

_tablename:نام جدولی که رکوردها در آن قرار داده شده اند را مشخص میکند و یک پارامتر ورودی می باشد.

_ fields:نام فیلد هایی از رکورد است و اگر * نوشته شود تمامی فیلد ها خوانده می شود و نوع آن ورودی می باشد.

_conditionalfield:شرط تعیین رکوردها را انتخاب می کند که می تواند نام دو یا چند فیلد باشداگر مقدارد هی به این پارامتر داده شود شرط در نظر گرفته نمی شود و تمامی رکوردهای جدول خوانده می شوند.

_ds:یک Data setاست که رکوردهای انتخاب شده در آن قرار می گیرند و نوع آن خروجی می باشد.

conditional compare operator_:عملگر مقایسه ی شرط است که می توانـد هـر یـک از عملگرهـای مقایسه ایی در sql باشد.مانند <= ،=>،<>،like و...

conditional logic operator_:نام یک عملگر منطقی مانند and وor است که فقط زمانی کـاربرد دارد که فیلد شرط بیشر از یک فیلد باشد ونوع آن ورودی است.

conditional value_:مقادیر را برای فیلد های شرطی تعیین می کند.

conditional_:شرط به روز رسانی می باشد .یعنی رکوردهایی که این شـرط در مـورد آنهـا صـدق مـی کندبه روز رسانی می شوند.

تابع بعدی برای پاک کردن یک یا تعدادی از رکوردها می باشد.

```
//-- Select record from database and puts in dataset
public void Delete_Record(string _TableName, string _ConditionalFileds, string
_ConditionalCompareOperator, string _ConditionalLogicOperator, params string[] _ConditionalValues)
{
string cmdtext = "";
if (_ConditionalFileds != string.Empty)
{
cmdtext = "DELETE FROM " + _TableName + " WHERE ";
string prm = "";
string[] Fields = new string[100];
int i = 0;
foreach (char c in _ConditionalFileds)
{
if (c != ',')
Fields[i] += c;
else
i++;
}
i = 0;
Command.Parameters.Clear();
foreach (string Value in _ConditionalValues)
{
prm = "@Val_" + i.ToString();
Command.Parameters.Add(prm, SqlDbType.NVarChar);
Command.Parameters[prm].Value = Value.Trim();
cmdtext += Fields[i] + " " + _ConditionalCompareOperator + " " + prm + " " + _ConditionalLogicOperator
+ " ";
i++;
}
cmdtext = cmdtext.Substring(0, cmdtext.Length - (_ConditionalLogicOperator.Length + 2));
}
else
cmdtext = "DELETE FROM " + _TableName;
ExecuteSQLCommand(cmdtext);
}
```

تا بع delete برای پاک کردن یک یا تعدادی از رکوردها با ارایه ی شرط خاصی می باشد.
پارامتر های این تابع عبارتند از:
_tablename:نام جدولی که رکوردها در آن قرار دهده شده اند را مشخص میکند و یک پارامتر ورودی می باشد.
_ fields:نام فیلد هایی از رکورد است و اگر * نوشته شود تمامی فیلد ها خوانده می شود و نوع آن ورودی می باشد.
_conditionalfield:شرط تعیین رکوردها را انتخاب می کند که می تواند نام دو یا چند فیلد باشداگر مقدار هی به این پارامتر داده شود شرط در نظر گرفته نمی شود و تمامی رکوردهای جدول خوانده می شوند.
_ds:یک Data setاست که رکوردهای انتخاب شده در آن قرار می گیرند و نوع آن خروجی می باشد.
_conditional compare operator:عملگر مقایسه ی شرط است که می تواند هر یک از عملگرهای مقایسه ایی در sql باشد.مانند <= ،=>،<>،like و...
_conditional logic operator:نام یک عملگر منطقی مانند and وor است که فقط زمانی کاربرد دارد که فیلد شرط بیشر از یک فیلد باشد ونوع آن ورودی است.
_conditional value:مقادیر را برای فیلد های شرطی تعیین می کند.
در این تابع نیز اگر رکوردهایی را با شرط خاصی بخواهیم حذف کنیم از این قسمت برنامه استفاده می کنیم:

```
if (_ConditionalFileds != string.Empty)
{
cmdtext = "DELETE FROM " + _TableName + " WHERE ";
string prm = "";
string[] Fields = new string[100];
int i = 0;
foreach (char c in _ConditionalFileds)
{
if (c != ',')
Fields[i] += c;
else
i++;
}
i = 0;
Command.Parameters.Clear();
foreach (string Value in _ConditionalValues)
{
prm = "@Val_" + i.ToString();
Command.Parameters.Add(prm, SqlDbType.NVarChar);
Command.Parameters[prm].Value = Value.Trim();
cmdtext += Fields[i] + " " + _ConditionalCompareOperator + " " + prm + " " + _ConditionalLogicOperator
+ " ";
i++;
}
```

```
cmdtext = cmdtext.Substring(0, cmdtext.Length - (_ConditionalLogicOperator.Length + 2));
}
```

این دستور sql به ما می گوید که رکوردی را از جدول حذف کن که دارای شرط یا شرایط خاصی باشد .

اگر شرطی برای حذف رکوردها نداشته باشیم یا بخواهیم کل رکوردها را حذف کنیم از قسمت else آن استفاده می کنیم.

قسمت بعدی برنامه مربوط به تابع encode می باشد:

```
//-- Permission EnCode
public void PermitEncode(char[] _PermitText, out int _PCode)
{

int code = 0;
string s = _PermitText[0].ToString() + _PermitText[1].ToString() + _PermitText[2].ToString() +
_PermitText[3].ToString();
switch (s)
{
case "0000": code = 0; break;
case "0001": code = 1; break;
case "0010": code = 2; break;
case "0011": code = 3; break;
case "0100": code = 4; break;
case "0101": code = 5; break;
case "0110": code = 6; break;
case "0111": code = 7; break;
case "1000": code = 8; break;
case "1001": code = 9; break;
case "1010": code = 10; break;
case "1011": code = 11; break;
case "1100": code = 12; break;
case "1101": code = 13; break;
case "1110": code = 14; break;
case "1111": code = 15; break;
}
_PCode = code;
}
```

هر کاربری دارای یک مجوز دسترسی می باشد که بدین وسیله هر کس بتواند با مجوزی که از مدیر می گیرد از نرم افزار استفاده کند.

این مجوز ها برای بالا بردن سطح امنیت می باشند.در غرفه ی کاربران در برنامه ی "adminestator"
ما ٤ مجوز دسترسی داریم که مختص پرسنل فروشگاه می باشند و با علامتگذاری هر کدام از آنها یک مجوز دسترسی به پرسنل داده می شود.
ورودی آن رشته ی حاوی کد مجوز و خروجی آن عدد مجوز می باشد.

```
string s = _PermitText[0].ToString() + _PermitText[1].ToString() + _PermitText[2].ToString() +
_PermitText[3].ToString();
```

در این دستور تک تک کدهای مجوز را به صورت رشته دریافت میکنیم و سپس با یک دستور swich کد این رشته را پیدا می کنیم.

این تابع عکس تابع encode می باشد:

```
//-- Permission DCode
public void PermitDcode(int _PCode,out char[] _PermitText)
{
string code = "";
switch (_PCode)
{
case 0: code = "0000"; break;
case 1: code = "0001"; break;
case 2: code = "0010"; break;
case 3: code = "0011"; break;
case 4: code = "0100"; break;
case 5: code = "0101"; break;
case 6: code = "0110"; break;
case 7: code = "0111"; break;
case 8: code = "1000"; break;
case 9: code = "1001"; break;
case 10: code = "1010"; break;
case 11: code = "1011"; break;
case 12: code = "1100"; break;
case 13: code = "1101"; break;
case 14: code = "1110"; break;
case 15: code = "1111"; break;
}
_PermitText = code.ToCharArray();
}
```

این تابع برای برگرداندن تاریخ می باشد:

```
public string showdate(DateTime date, Boolean shortdate)
{
string[] week = { "شنبه", "یکشنبه", "دوشنبه", "شنبه سه", "چهارشنبه", "شنبه پنج", "جمعه" };
string[] months = { "فروردین", "اردیبهشت", "خرداد", "تیر", "مرداد", "شهریور", "مهر", "آبان", "آذر", "دی",
"بهمن", "اسفند" };
short d = 0;
DateTime a = date;
DayOfWeek tempdayofweek = a.DayOfWeek;
switch (tempdayofweek)
{
case DayOfWeek.Saturday: d = 0; break;
case DayOfWeek.Sunday: d = 1; break;
case DayOfWeek.Monday: d = 2; break;
case DayOfWeek.Tuesday: d = 3; break;
case DayOfWeek.Wednesday: d = 4; break;
case DayOfWeek.Thursday: d = 5; break;
case DayOfWeek.Friday: d = 6; break;
}


int day = int.Parse(a.Day.ToString());
int month = int.Parse(a.Month.ToString());
int year = int.Parse(a.Year.ToString());

year = (year == 0) ? 2000 : year;
if (year < 1000)
{ year += 2000;/*:true;*/}

year -= ((month < 3) || ((month == 3) && (day < 21))) ? 622 : 621;

switch (month)
{
case 1: if (day < 21) { month = 10; day += 10; } else { month = 11; day -= 20; } break;
case 2: if (day < 20) { month = 11; day += 11; } else { month = 12; day -= 19; } break;
case 3: if (day < 21) { month = 12; day += 9; } else { month = 1; day -= 20; } break;
case 4: if (day < 21) { month = 1; day += 11; } else { month = 2; day -= 20; } break;
case 5:
case 6: if (day < 22) { month -= 3; day += 10; } else { month -= 2; day -= 21; } break;
case 7:
case 8:
case 9: if (day < 23) { month -= 3; day += 9; } else { month -= 2; day -= 22; } break;
case 10: if (day < 23) { month = 7; day += 8; } else { month = 8; day -= 22; } break;
case 11:
case 12: if (day < 22) { month -= 3; day += 9; } else { month -= 2; day -= 21; } break;
default: break;
}
if (shortdate != true)
{
return (week[d] + " " + day + " " + months[month - 1] + " " + year);
}
else
{
string smonth = "", sday = "";
if (month >= 1 && month <= 9)
{
smonth = "0" + month.ToString();
```

```
}
else
{
smonth = month.ToString();
}

if (day >= 1 && day <= 9)
{
sday = "0" + day.ToString();
}
else
{
sday = day.ToString();
}
return (year + "/" + smonth + "/" + sday);
}
}
}
}
```

این تابع یک تاریخ میلادی را گرفته و به تاریخ شمسی تبدیل می کندو تاریخ شمسی را بـه صـورت رشـته برمی گرداند.

پارامترها:

Date:یک پارامتر ورودی است که حاوی تاریخ میلادی می باشد.

Shortdate:دارای دو وضعیت true و false می باشد.

اگر د ر وضعیت true باشد:خروجی تابع به صورت روز/ماه/سال است مانند ۸٦/٦/۱۸

اگر در وضعیت false باشد:خروجی تابع به صورت نام روز عدد روز نام ماه عـدد سـال مـی باشـد ماننـد دوشنبه ۱۹ شهریور ۱۳۸٦.

همانطور که قبلا گفته شده اقلام موجود در این فروشگاه بـه ٤ قسـمت کتاب،رسـانه ی آموزشـی ، لـوازم التحریر و انواع پوستر و نقشه تقسیم می شود.

اکنون نگاهی اجمالی داریم به فرمهای مربوط به اقلام فروشگاه:

فرم مربوط به غرفه ی کتاب در برنامه ی مدیر:

```
public partial class Form_Book : Form
{
IRA ira = new IRA();
DataSet DS = new DataSet();

public Form_Book()
{
InitializeComponent();
}
```

تابع مربوط به بارگذاری رکوردهادر لیست:

پارامتر ورودی این تابع شماره ی گروه می باشد

```
private void bind(string _GROUPID)
{
ira.Select_Record("BOOK", "ID, NAME, WRITER, COMPILER, DEVOLEPER, PRINTYEAR, _COUNT,
PRICE", out DS, "GID", "=", "AND", _GROUPID);
dataGridView_BookList.DataSource = DS;
dataGridView_BookList.DataMember = DS.Tables[0].ToString();
}
```

در این قسمت نیز به دلیل استفاده از پایگاه داده ابتدا به آن متصل شده ایم و یک dataset نیز برای نگه داشتن جواب ها تعریف کرده ایم.
کاری که تابع bind برای ما انجام می دهد این است که در قالب یک دستور sql رکورد یا رکوردهایی را که با شرط ما همخوانی دارند را برای ما به نمایش در می آورد.

در دستور :

```
ira.Select_Record("BOOK", "ID, NAME, WRITER, COMPILER, DEVOLEPER, PRINTYEAR, _COUNT,
PRICE", out DS, "GID", "=", "AND", _GROUPID);
```

به پایگاه داده می گوییم که کلیه فیلد های یک رکورد که شامل نام ، شماره کتاب ،نویسنده ،ناشر ، قیمت ،تعداد و...را به نمایش در بیاور که شماره ی گروه آن با چیزی که به عنوان ورودی وارد این دستور شده است یکی باشد.
و and در اینجا مفهومی ندارد برای اینکه فیلدی که باید برای عملگر and در اینجا وجود داشه باشد تعریف نشده است.

تابع مربوط به جستجو:

```
private void Filter(string _Filed, string _Operator, string _Value)
{
SqlConnection Connection = new SqlConnection("Data Source=localhost;Initial Catalog=Shop;Integrated Security=True");
SqlCommand Command = new SqlCommand();
SqlDataAdapter Adapter = new SqlDataAdapter();
DS.Clear();
Command.CommandText = "SELECT ID, NAME, WRITER, COMPILER, DEVOLEPER, PRINTYEAR, _COUNT, PRICE FROM BOOK WHERE GID='" + comboBox_Group.SelectedValue.ToString() + "' AND " + _Filed + _Operator +  _Value ;
Command.Connection = Connection;
Connection.Open();
Adapter.SelectCommand = Command;
Adapter.Fill(DS);
Connection.Close();
}
```

در ابتدای تابع فیلتر یک connection تعریف کرده ایم که با توجه به آن میبینیم که "datasource=localhost" قرار داده ایم و این به این خاطر می باشد که می خواهیم این کار ها بر روی سیستم فعلی یعنی سیستمی که هم اکنون در حال استفاده می باشد انجام شود و اگر فروشگاه به شبکه وصل شود با توجه به مشکلات امنیتی و اینکه هر کاربری با دستکاری کردن این داده مشگل ناسازگاری به وجود نیاورد استفاده شده است.

در خط بعدی یک command تعریف کرده ایم برای اجرای دستورات sql و در دستور بعدی یک adpter تعریف شده برای خواندن رکوردها از بانک اطلاعاتی.

در دستور بعدی ds را پاک میکنیم تا اگر از اطلاعات قبلی در آن موجود باشد بر روند کار اثری نداشته باشد .

تابع فیلتر برای جستجو کردت طراحی شده است و به کمک combobox هایی که در سمت راست طراحی شده است این کار انجام می پذیرد.

در زمانی که comnobox های مربط به جستجو تغییر می کنند این نابع فراخوانی می شود.

کد مربوط به اتصال combobox به بانک برای آوردن لیست:

```
private void Form_MangeBook_Load(object
sender, EventArgs e)
{
// TODO: This line of code loads data into the 'shopDataSet_Group._GROUP' table. You can move, or
remove it, as needed.
this._GROUPTableAdapter.FillBy(this.shopDataSet_Group._GROUP, "01");
bind(comboBox_Group.SelectedValue.ToString());
comboBox_CountFilter.SelectedIndex = 0;
comboBox_IDFilter.SelectedIndex = 0;
comboBox_NameFilter.SelectedIndex = 0;
comboBox_DevoleperFilter.SelectedIndex = 0;
comboBox_WriterFilter.SelectedIndex = 0;
comboBox_CompilerFilter.SelectedIndex = 0;
comboBox_PriceFilter.SelectedIndex = 0;
comboBox_PrintYearFilter.SelectedIndex = 0;
}
```

ودر انتها انتخاب اولین آیتم در combobox های مربوط به جستجو.

```
private void button_Back_Click(object sender, EventArgs e)
{
this.Close();
}
```

این تابع برای برگشتن به پنجره ی قبلی می باشد .به این صورت که با این دستور کوتاه گفته شده که فرمی که اکنون فرم فعال است را ببند.

تابع مربوط به ثبت اطلاعات:

```
private void button_Save_Click(object sender, EventArgs e)
{
if (textBox_ID.Text == string.Empty || textBox_Name.Text == string.Empty || textBox_Price.Text ==
string.Empty || textBox_Count.Text == string.Empty)
{
label_Error.Show();
return;
}
```

در هنگام ثبت اطلاعات باید تمامی اطلاعات مربوط به کالا را به جز توضیحات وارد کرد.

در این قسمت برنامه چک می کند که آیا شماره یا نام یا قیمت و تعداد کتاب ها وارد نشده است و تهی می باشد؟اگر جواب درست باشد پیغام خطا می دهد و به برنامه فراخوان باز می گردد.
اگر شرط بالا اجرا نشود به این معنا خواهد بود که مشخصات کامل درج شده است.لذا هم اکنون باید جستجو شود که آیا این کتاب وجود دارد یا برای اولین بار وارد فروشگاه می شود.در صورت درست بودن قسمت اول باید به روز رسانی و برای درست بودن قسمت دوم باید عملیات درج انجام شود.

```
ira.Select_Record("BOOK", "ID, NAME, WRITER, COMPILER, DEVOLEPER, PRINTYEAR, _COUNT,
PRICE", out DS, "GID, ID", "=", "AND", comboBox_Group.SelectedValue.ToString(), textBox_ID.Text);
int _Count = 0;
if (DS.Tables[0].Rows.Count > 0)
{
_Count = int32.Parse(DS.Tables[0].Rows[0]["_COUNT"].ToString());
_Count += Int32.Parse(textBox_Count.Text);
ira.Update_Record("BOOK", "GID='" + comboBox_Group.SelectedValue.ToString() + "' AND ID='" +
textBox_ID.Text.Trim() + "'", "_COUNT, PRICE", _Count.ToString(), textBox_Price.Text);

}
```

در این قسمت برنامه کتابی را پیدا می کند که شماره ی کتاب و شماره ی گروه آن با چیزی که ما می خواهیم مطابقت دارد و چک می کند که آیا از این کتاب باقی مانده است یا نه؟اگر از این کتاب وجود داشته باشد تعداد جدید را به تعداد قبلی اضافه می کند و سپس فیلد های قیمت و تعداد را به روز رسانی می کند.

```
else
ira.Insert_Record("BOOK", "GID, ID, NAME, WRITER, COMPILER, DEVOLEPER, PRINTYEAR,
PRICE, _COUNT, NOTE", comboBox_Group.SelectedValue.ToString(), textBox_ID.Text,
textBox_Name.Text, textBox_Writer.Text, textBox_Compiler.Text, textBox_Devoleper.Text,
textBox_PrintYear.Text, textBox_Price.Text, textBox_Count.Text, textBox_Note.Text);
button_Clear_Click(sender, e);
bind(comboBox_Group.SelectedValue.ToString());

}
```

اگر کتاب وارد شده به فروشگاه از قبل در فروشگاه وجود نداشت عملیات درج انجام می شود و سپس برای نمایش کتاب در gridview تابع bind فراخوانی می شود.

```
private void textBox_Price_KeyPress(object sender, KeyPressEventArgs e)
{
if ((e.KeyChar < '0' || e.KeyChar > '9') && e.KeyChar != '\b')
e.KeyChar = '\0';
}
```

در این قسمت برنامه چک می شود که برای تعداد از حروف الفبا استفاده نکنیم و یا بـه عبـارتی بهتـر اگـر چیزی غیر از عدد وارد شده آن را قبول نکند.در اینجا backspace هم قابل قبول است تا کـاربر بتوانـد اشتباه خود را تصحیح کند و آن را پاک کند.

کد مربوط به وارد کردن قیمت:

```
private void textBox_Price_TextChanged(object sender, EventArgs e)
{
foreach(char c in (sender as TextBox).Text)
if (c < '0' || c > '9')
{
(sender as TextBox).Text = "0";
break;
}
if ((sender as TextBox).Text == string.Empty) (sender as TextBox).Text = "0";
}
```

در این قسمت برنامه چک می شود که کاربر برای وارد کردن حروف از عملیات کپی استفاده نکند .
اگر کاربر از کپی استفاده کرد و این کپی هم دارای حروفی به جز عدد بود مثل قسمت بالا این حـروف قبول نمی شوند.
این قسمت از برنامه برای امنیت بیشتر می باشد.

```
private void button_Help_Click(object sender, EventArgs e)
{
Form_BookHelp Help = new Form_BookHelp();
Help.ShowDialog();
    }
```

اگر کاربر از دکمه ی help برای استفاده از نرم افزار استفاده کرد این قسمت برنامه فراخوانی می شود.

این تابع هنگامی که در textbox شماره کلید enter زده شود فراخوانی می شود:

```
private void Check()
{
if (textBox_ID.Text == string.Empty)
{
bind(comboBox_Group.SelectedValue.ToString());
return;
}
ira.Select_Record("BOOK", "*", out DS, "GID, ID", "=", "AND",
comboBox_Group.SelectedValue.ToString(), textBox_ID.Text);
if (DS.Tables[0].Rows.Count > 0)
{
textBox_Name.Text = DS.Tables[0].Rows[0]["NAME"].ToString();
textBox_Writer.Text = DS.Tables[0].Rows[0]["WRITER"].ToString();
textBox_Compiler.Text = DS.Tables[0].Rows[0]["COMPILER"].ToString();
textBox_Devoleper.Text = DS.Tables[0].Rows[0]["DEVOLEPER"].ToString();
textBox_PrintYear.Text = DS.Tables[0].Rows[0]["PRINTYEAR"].ToString();
textBox_Count.Text = DS.Tables[0].Rows[0]["_COUNT"].ToString();
textBox_Price.Text = DS.Tables[0].Rows[0]["PRICE"].ToString();
textBox_Note.Text = DS.Tables[0].Rows[0]["NOTE"].ToString();
}
```

در این قسمت برنامه دردستور sql گفته شده است که در جدول کتاب همه ی آنهایی را پیدا کند که شماره ی آن با چیزی که در text box نوشته شده مطابقت داشته باشد و سپس کلید enter زده شده باشد..سپس چک می کند که آیا در این جستجو چیزی یافت شده است یا خیر .که اگر تعداد بیشتر از صفر باشد یعنی این کتاب وجود داشته و در این صورت به textbox هایی که وجود دارد برار با مشخصات آن کتاب می شود و اگر تعداد برابر صفر بود یعنی از ان نوع کتاب وجود نداشته است لذا textbox های مربوطه تهی می شوند.این قسمت برنامه در دستور Else انجام می شود.

```
else
{
textBox_Name.Text = string.Empty;
textBox_Writer.Text = string.Empty;
textBox_Compiler.Text = string.Empty;
textBox_Devoleper.Text = string.Empty;
textBox_PrintYear.Text = string.Empty;
textBox_Note.Text = string.Empty;
textBox_Price.Text = string.Empty;
textBox_Count.Text = string.Empty;
}
textBox_ID.SelectAll();
}
```

کد مربوط به قسمت پاک کردن اطلاعات:

```
private void button_Clear_Click(object sender, EventArgs e)
{
textBox_ID.Text = string.Empty;
textBox_Name.Text = string.Empty;
textBox_Writer.Text = string.Empty;
textBox_Compiler.Text = string.Empty;
textBox_Devoleper.Text = string.Empty;
textBox_PrintYear.Text = string.Empty;
textBox_Note.Text = string.Empty;
textBox_Price.Text = string.Empty;
textBox_Count.Text = string.Empty;
textBox_ID.Focus();
}
```

این قسمت عملکرد دکمه ی clear را نشان می دهد که در آن تمامی Textbox ها تهی می شوند.

تابع مربوط به فراخوانی تابع check:

```
private void textBox_ID_KeyPress(object sender, KeyPressEventArgs e)
{

if ( e.KeyChar == '\r') Check();
}
```

در این قسمت گفته شده است که اگر بعد از رشته ایی کـه وارد شـد کلیـد enter زده شـده اسـت تـابع check فراخوانی شود.

این قسمت برنامه برای پاک کزدن کالا به کمک link "حذف شود"در اولین سلول datagridview می باشد.

```
private void dataGridView_BookList_CellClick(object sender, DataGridViewCellEventArgs e)
{
if (e.ColumnIndex == 0)
{
string s = "کتاب " + dataGridView_BookList.Rows[e.RowIndex].Cells[2].Value.ToString() + " حذف؟شود";
if (MessageBox.Show(s, "مدیریت", MessageBoxButtons.YesNo, MessageBoxIcon.Question,
MessageBoxDefaultButton.Button1, MessageBoxOptions.RtlReading) == DialogResult.Yes)
{
ira.Delete_Record("BOOK", "GID, ID", "=", "AND", comboBox_Group.SelectedValue.ToString(),
dataGridView_BookList.Rows[e.RowIndex].Cells[1].Value.ToString());
bind(comboBox_Group.SelectedValue.ToString());
}
```

```
}
}
```

در این قسمت برنامه اگر label حذف شود را انتخاب کنیم در ابتدا یک پیغام می دهد که آیا کتاب انتخاب شده حذف شود؟بعد از قبول کردن این گزینه کتاب مورد نظر ذف شده و در انتها دوباره تابع Bind فراخوانی می شود.

این قسمت برنامه برای جستجو بر طبق تعداد می باشد:

```
private void comboBox_CountFilter_SelectedIndexChanged(object sender, EventArgs e)
{
if (comboBox_CountFilter.SelectedIndex == 0)
bind(comboBox_Group.SelectedValue.ToString());
else
Filter("_COUNT", comboBox_CountFilter.Text, textBox_Count.Text);
}
```

در این قسمت برنامه تابع فیلتر فراخوانی شده است به این ترتیب که در ابتدا چک می شود که کدام گزینه برای جستجو انتخاب شده است.اگر انتخاب بر روی index شماره ی صفر باشد یعنی گزینه ی همه انتخاب شده است و در این صورت تابع bind فراخوانی می شود.
در غیر این صورت با توجه به گزینه ی انتخاب شده تابع filter فراخوانی می شود.

این قسمت برنامه برای جستجو بر طبق قیمت می باشد:

```
private void comboBox_Price_SelectedIndexChanged(object sender, EventArgs e)
{
if (comboBox_PriceFilter.SelectedIndex == 0)
bind(comboBox_Group.SelectedValue.ToString());
else
Filter("PRICE", comboBox_PriceFilter.Text, textBox_Price.Text);
}
```

در این قسمت برنامه تابع فیلتر فراخوانی شده است به این ترتیب که در ابتدا چک می شود که کدام گزینه برای جستجو انتخاب شده است.اگر انتخاب بر روی index شماره ی صفر باشد یعنی گزینه ی همه انتخاب شده است و در این صورت تابع bind فراخوانی می شود.
در غیر این صورت با توجه به گزینه ی انتخاب شده تابع filter فراخوانی می شود.

این قسمت برنامه برای جستجو بر طبق نام کتاب می باشد:

```
private void comboBox_Title_SelectedIndexChanged(object sender, EventArgs e)
{
if (textBox_Name.Text == string.Empty) bind(comboBox_Group.SelectedValue.ToString());
switch (comboBox_NameFilter.SelectedIndex)
{
case 0: bind(comboBox_Group.SelectedValue.ToString()); break;
case 1: Filter("NAME", " LIKE  N", "'%" + textBox_Name.Text + "%'"); break;
case 2: Filter("NAME", "= N'", textBox_Name.Text+"'"); break;
}
}
```

در یک دستور swich بررسی شده که جستجو بر چه اساسی انجام شود .اگر گزینه ی همه انتخاب شود تابع bind فراخوانی می شود.اگر گزینه ی شبیه انخاب شود همه کتاب هایی را نمایش می دهد که نام آن شبیه برچیزی باشد که مد نظر ما می باشد و اگر برابر انتخاب شود همه ی کتاب هایی انتخاب می شود که نام آن برابر با چیزی باشد که مد نظر ماست.

جستجو بر اساس شماره ی کتاب:

```
private void comboBox_IDFilter_SelectedIndexChanged(object sender, EventArgs e)
{
if (textBox_ID.Text == string.Empty) bind(comboBox_Group.SelectedValue.ToString());
switch (comboBox_IDFilter.SelectedIndex)
{
case 0: bind(comboBox_Group.SelectedValue.ToString()); break;
case 1: Filter("ID", " LIKE  N", "'%" + textBox_ID.Text + "%'"); break;
case 2: Filter("ID", "= N'", textBox_ID.Text+"'"); break;
}
}
```

در یک دستور swich بررسی شده که جستجو بر چه اساسی انجام شود .اگر گزینه ی همه انتخاب شود تابع bind فراخوانی می شود.اگر گزینه ی شبیه انخاب شود همه کتاب هایی را نمایش می دهد که شماره ی آن شبیه برچیزی باشد که مد نظر ما می باشد و اگر برابر انتخاب شود همه ی کتاب هایی انتخاب می شود که شماره ی آن برابر با چیزی باشد که مد نظر ماست.

جستجو بر اساس نویسنده:

```
private void comboBox_WriterFilter_SelectedIndexChanged(object sender, EventArgs e)
{
if (textBox_Writer.Text == string.Empty) bind(comboBox_Group.SelectedValue.ToString());
switch (comboBox_WriterFilter.SelectedIndex)
{
case 0: bind(comboBox_Group.SelectedValue.ToString()); break;
case 1: Filter("WRITER", " LIKE  N", "'%" + textBox_Writer.Text + "%'"); break;
case 2: Filter("WRITER", "= N'", textBox_Writer.Text+"'"); break;
}
}
```

در یک دستور swich بررسی شده که جستجو بر چه اساسی انجام شود .اگر گزینه ی همه انتخاب شود تابع bind فراخوانی می شود.اگر گزینه ی شبیه انخاب شود همه کتاب هایی را نمایش می دهد که نویسنده ی آن شبیه برچیزی باشد که مد نظر ما می باشد و اگر برابر انتخاب شود همه ی کتاب هایی انتخاب می شود که نویسنده ی آن برابر با چیزی باشد که مد نظر ماست.

جستجو بر اساس مترجم:

```
private void comboBox_CompilerFilter_SelectedIndexChanged(object sender, EventArgs e)
{
if (textBox_Compiler.Text == string.Empty) bind(comboBox_Group.SelectedValue.ToString());
switch (comboBox_CompilerFilter.SelectedIndex)
{
case 0: bind(comboBox_Group.SelectedValue.ToString()); break;
case 1: Filter("COMPILER", " LIKE  N'", "%" + textBox_Compiler.Text + "%'"); break;
case 2: Filter("COMPILER", "= N'", textBox_Compiler.Text+"'"); break;
}
}
```

در یک دستور swich بررسی شده که جستجو بر چه اساسی انجام شود .اگر گزینه ی همه انتخاب شود تابع bind فراخوانی می شود.اگر گزینه ی شبیه انخاب شود همه کتاب هایی را نمایش می دهد که مترجم آن شبیه برچیزی باشد که مد نظر ما می باشد و اگر برابر انتخاب شود همه ی کتاب هایی انتخاب می شود که مترجم آن برابر با چیزی باشد که مد نظر ماست.

جستجو بر اساس ناشر:

```
private void comboBox_DevoleperFilter_SelectedIndexChanged(object sender, EventArgs e)
{
if (textBox_Devoleper.Text == string.Empty) bind(comboBox_Group.SelectedValue.ToString());
switch (comboBox_DevoleperFilter.SelectedIndex)
{
case 0: bind(comboBox_Group.SelectedValue.ToString()); break;
case 1: Filter("DEVOLEPER", " LIKE  N'", "%" + textBox_Devoleper.Text + "%'"); break;
case 2: Filter("DEVOLEPER", "= N'", textBox_Devoleper.Text+"'"); break;
}
}
```

در یک دستور swich بررسی شده که جستجو بر چه اساسی انجام شود .اگر گزینه ی همه انتخاب شود تابع bind فراخوانی می شود.اگر گزینه ی شبیه انخاب شود همه کتاب هایی را نمایش می دهد که ناشر آن شبیه برچیزی باشد که مد نظر ما می باشد و اگر برابر انتخاب شود همه ی کتاب هایی انتخاب می شود که ناشر آن برابر با چیزی باشد که مد نظر ماست.

جستجو بر اساس تاریخ چاپ:

```
private void comboBox_PrintYearFilter_SelectedIndexChanged(object sender, EventArgs e)
{
if (textBox_PrintYear.Text == string.Empty) bind(comboBox_Group.SelectedValue.ToString());
switch (comboBox_PrintYearFilter.SelectedIndex)
{
case 0: bind(comboBox_Group.SelectedValue.ToString()); break;
case 1: Filter("PRINTYEAR", " LIKE  N'", "%" + textBox_PrintYear.Text + "%'"); break;
case 2: Filter("PRINTYEAR", "= N'", textBox_PrintYear.Text+"'"); break;
}
}
```

در یک دستور swich بررسی شده که جستجو بر چه اساسی انجام شود .اگر گزینه ی همه انتخاب شود تابع bind فراخوانی می شود.اگر گزینه ی شبیه انخاب شود همه کتاب هایی را نمایش می دهد که تاریخ چاپ آن شبیه برچیزی باشد که مد نظر ما می باشد و اگر برابر انتخاب شود همه ی کتاب هایی انتخاب می شود که تاریخ چاپ آن برابر با چیزی باشد که مد نظر ماست.

فرم بعد مربوط به فرم مدیریت انبار می باشد در این فرم یک متغیر به نام "shopid" داریم که ابتدا با مقدار ۰۱ مقدار دهی شده است.سپس در یک دستور Swich تعیین می کنیم که اگر مقدار ۰۱ در این متغیر باقی ماند منظور همان فرم مربوط به کتاب می باشد.

اگر این مقدار با ۰۲ عوض شد این عدد کد مربوط به رسانه ی آموزشی ی باشد.

اگر این مقدار با ۰۳ عوض شد این عدد کد مربوط به پوستر می باشد.

اگر این عدد با ۰٤ عوض شد این عدد کد مربوط به لوازم التحریر میباشد.

فرم مربوط به مدیریت انبار:

```
namespace BookShop
{
public partial class Form_FileManage : Form
{
string ShopID = "01";
public Form_FileManage()
{
InitializeComponent();
}

private void button_Back_Click(object sender, EventArgs e)
{
this.Close();
}

private void radioButton_Book_CheckedChanged(object sender, EventArgs e)
{
ShopID = (sender as RadioButton).Tag.ToString();
}

private void button_Go_Click(object sender, EventArgs e)
{
switch (ShopID)
```

```
{
case "01": Form_Book Book = new Form_Book(); Book.ShowDialog(); break;
case "02": Form_LearningMedia LearningMedia = new Form_LearningMedia();
LearningMedia.ShowDialog(); break;
case "03": Form_Pooster Pooster = new Form_Pooster(); Pooster.ShowDialog(); break;
case "04": Form_TypeWrite TypeWrite = new Form_TypeWrite(); TypeWrite.ShowDialog(); break;

}
}
}
}
```

اکنون نگاهی می اندازیم به فرم اصلی برنامه مدیر:

در ابتدا کلاس ira را معرفی می کنیم و سپس یک dataset جدید باز می کنیم تا اطلاعات مربوط به فرم در آن ریخته شود.در ابتدا تنها دکمه ی که فعال و قابل مشاهده می باشد دکمه ی مربوط به خـروج مـی باشد.کاربر تازه وارد برای خروج از نرم افزار می تواند از کلیک کردن بر روی این دکمه استفاده کند.

```
public partial class Form_Main : Form
{
IRA ira = new IRA();
DataSet DS = new DataSet();

public Form_Main()
{
InitializeComponent();
}

private void Form_Main_FormClosed(object sender, FormClosedEventArgs e)
{
Application.Exit();
}

private void button_Close_Click(object sender, EventArgs e)
{
this.Close();
}

private void button_Shops_Click(object sender, EventArgs e)
{
Form_Shops Frm = new Form_Shops();
Frm.ShowDialog();
```

```
}

private void button_Users_Click(object sender, EventArgs e)
{
Form_Users Frm = new Form_Users();
Frm.ShowDialog();
}

private void button_File_Click(object sender, EventArgs e)
{
Form_FileManage Frm = new Form_FileManage();
Frm.ShowDialog();
}

private void linkLabel_Login_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
label_LoginFailed.Hide();
if (textBox_UserID.Text == string.Empty || textBox_Pass.Text == string.Empty) return;
ira.Select_Record("ACCOUNT", "NAME, PERMISSION", out DS, "ID, PASSWORD", "=", "AND",
textBox_UserID.Text, textBox_Pass.Text);
if (DS.Tables[0].Rows.Count < 1)
{
label_LoginFailed.Show();
label_Weelcom.Hide();
textBox_Pass.Clear();
textBox_UserID.Clear();
textBox_UserID.Focus();
return;
}
```

در دستور بعد از فراخوانی ira گفه شده که اگر کاربر دکمه ی خروج را زد کل Application بسته شود و اگر label"خروج از سیستم" را زد از صفحه ی خود خارج شده و به فرم اول برنامه ی خریدار باط گردد.

کاربر برای ورود به سیستم باید نام کاربری و رمز عبور خود را وارد کند.کاربرانی که می توانند وارد این نرم افزار شوند شامل مدیر،پرسنل و خریداران می باشند .اما هر کدام مجوز دسترسی مربوط به خود را دارا می باشند.مثلا مدیر مجوز دسترسی به همه ی قسمت ها را دارد اما خریدار هیچ گونه مجوز دسترسی ندارد.در تابع فوق گفته شده است که اگر نام کاربری یا رمز عبور وارد نشده است به آن تریب اثر داده نشود.

سپس با یک دستور sql گفته شده که از جدول Account نام و رمز عبوری که وارد شده است را بررسی کند تا متوجه شود که این نام کاربری و رمز عبور وجود دارد یا خیر و در صورت وجود داشتن چه مجوز دسترسی هایی دارد.

اگر چنین نام و مشخصاتی یافت نشد پیغام خطا می دهد و نام کاربری و رمز عبور پاک می شود.

```
int Permission = Int32.Parse(DS.Tables[0].Rows[0]["PERMISSION"].ToString());
if (Permission == 0)
{
string msg =DS.Tables[0].Rows[0]["NAME"].ToString() + " ،" + "ندارید دسترسی مجوز شما.";
```

```
MessageBox.Show(msg, "مدیریت", MessageBoxButtons.OK, MessageBoxIcon.Exclamation,
MessageBoxDefaultButton.Button1, MessageBoxOptions.RtlReading);
return;
}
label_Weelcom.Show();
label_Weelcom.Text = " آمدید خوش " + DS.Tables[0].Rows[0]["NAME"].ToString();
```

مجوز دسترسی در این برنامه از عدد ۰ تا ۱۵ می باشد .این اعداد به صورت باینری هستند .به این صورت که مثلا باینری عدد صفر به صورت ۰۰۰۰ می باشد که این اعداد از راست به چپ مجوز دسترسی به انبار،حسابداری ، غرفه ها و کاربران می باشد.مثلا عدد باینری ۰۰۰۱ معرف آن است که این کاربر مجوز دسترسی به انبار را دارد.

همینطور که دیده می شود شخصی با مجوز دسترسی ۰ به هیچ کدام از غرفه ها دسترسی ندارد.لذا یک پیغام به وی نمایش داده می شود که شما هیچ گونه مجوز دسترسی ندارید.

و اگر مجوز دسترسی به یک یا چند غرفه را داشته باشد به وی خوش آمد گفته می شود و با فعال شدن غرفه هایی که کاربر مجوز دسترسی به آنها را دارد به وی هشدار داده می شود که چه مجوز هایی را دارا می باشد.

```
//-----------
if (Permission >= 0 && Permission <= 7)
button_Users.Enabled = false;
else
button_Users.Enabled = true;
```

همانطور که در قسمت فوق توضیح داده شد شخصی که وارد شده با مجوز دسترسی ۰ تا ۷ اجازه ی ورود به بخش کاربران را ندارد.

```
//------------

if ((Permission >= 0 && Permission <= 3) || (Permission >= 8 && Permission <= 11))
button_Shops.Enabled = false;
else
button_Shops.Enabled = true;
```

اگر عدد اعلام شده بین ۰ تا ۳ یا بین ۸ تا ۱۱ بود مجوز دسترسی به بخش غرفه ها را ندارد.

```
//-----------
```

```
if (Permission % 2 == 0)
button_File.Enabled = false;
else
button_File.Enabled = true;
```

عددی که به عنوان مجوز اعلام می شود اگر یک عدد زوج باشد مجوز دسترسی به انبار را ندارد.

```
//----------------
if (Permission == 0 || Permission == 1 || Permission == 4 || Permission == 5 || Permission == 8 || Permission ==
9 || Permission == 12 || Permission == 13)
button_Accounting.Enabled = false;
else
button_Accounting.Enabled = true;
textBox_Pass.Clear();
textBox_UserID.Clear();
textBox_UserID.Focus();
}
```

عددی که به عنوان مجوز وارد می شود اگر برابر با ۰ یا ۱ یا ٤ یا ٥ یا ۸ یا ۹ یا ۱۲ و یا ۱۳ باشد مجوز دسترسی به حسابداری را نخواهد داشت.

```
private void linkLabel_Logout_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
label_LoginFailed.Hide();
label_Weelcom.Hide();
button_Users.Enabled = false;
button_Shops.Enabled = false;
button_File.Enabled = false;
button_Accounting.Enabled = false;

}
```

این قسمت مربوط به دکمه ی خروج از سیستم می باشد. به این صورت که وقتی دکمه ی خروج از سیستم فعال شد.اگر پیغام خطا و یا خوش آمد گویی روی صفحه وجود دارد مخفی شده و مجوز های دسترسی به قسمت کاربران ، غرفه ها ، حسابداری و انبار غیر فعال شود.

کد مربوط به highlight کردن نام کاربری:

```
private void textBox_UserID_Enter(object sender, EventArgs e)
{
(sender as TextBox).SelectAll();
}
```

وقتی کاربر به قسمت userid رفت برای وارد کردن نام جدید،نامی که از قبل در آنجا نوشته شده بود به صورت highlight در آمده که پاک کردن آن برای کاربر راحت تر شود.

کد مربوط به ساعت:

```
private void timer1_Tick(object sender, EventArgs e)
{
label_Date.Text = "امروز: " + ira.showdate(DateTime.Today, false);
}
```

کد مربوط به نمایش قسمت حسابداری:

```
private void button_Accounting_Click(object sender, EventArgs e)
{
Form_AccountDlg AccountDlg = new Form_AccountDlg();
AccountDlg.ShowDialog();
}
}
```

فرم بعد مربوط به گزارش گیری در برنامه ی مدیر می باشد:

در ابتدا یک connection برای اتصال به پایگاه داده برقرار می کنیم و سپس یک Command برای اجرای دستورات و یک adapter برای خواندن رکوردها ایجاد می کنیم.

```
public partial class Form_Reports : Form
{
public Form_Reports()
{
InitializeComponent();
}
DataSet DS = new DataSet();
private void Report(string _FromDate, string _ToDate)
{
SqlConnection Connection = new SqlConnection("Data Source=localhost;Initial Catalog=Shop;Integrated
Security=True");
SqlCommand Command = new SqlCommand();
SqlDataAdapter Adapter = new SqlDataAdapter();

//------------------------
```

سپس ds را برای ریختن اطلاعات جدید پاک میکنیم.
در دستور sql می گوییم که مشخصات(نام ،قیمت ، سود ،تخفیف و...)کالایی را بده که تاریخ فروش آن از یک مقدار تا مقدار مشخص شده ایی باشد که تعریف شده است.سپس connection را باز کرده دستور را اجرا می کنیم و سپس آن را می بندیم.

```
DS.Clear();
Command.CommandText = "SELECT CLASSID, _DATE, CUSTOMER, NAME, _COUNT, PRICE,
SOOD, TAKHFIF FROM CUSTOMS WHERE _DATE >='" + _FromDate + "' AND _DATE <='" +
_ToDate + "'";
Command.Connection = Connection;
```

```
Connection.Open();
Adapter.SelectCommand = Command;
Command.ExecuteNonQuery();
Adapter.Fill(DS);
Connection.Close();
dataGridView_ReportList.DataSource = DS;
dataGridView_ReportList.DataMember = DS.Tables[0].ToString();
Int64 Sood = 0;
Int64 Sum = 0;
Int64 Takhfif = 0;
for (int i = 0; i <= DS.Tables[0].Rows.Count - 1; i++)
{
```

جمع کل فروش بدون تخفیف:

```
Sum                                    Int64.Parse(DS.Tables[0].Rows[i]["_COUNT"].ToString())                    *
Int64.Parse(DS.Tables[0].Rows[i]["PRICE"].ToString());
```

فرمول محاسبه ی سود:

```
  Sood += (Int64.Parse(DS.Tables[0].Rows[i]["SOOD"].ToString()) *
Int64.Parse(DS.Tables[0].Rows[i]["_COUNT"].ToString()) *
Int64.Parse(DS.Tables[0].Rows[i]["PRICE"].ToString())) / 100;
```

فرمول محاسبه ی جمع کل با تخفیف:

```
Takhfif += ((Int٦٤.Parse(DS.Tables[٠].Rows[i]["_COUNT"].ToString()) *
Int٦٤.Parse(DS.Tables[٠].Rows[i]["PRICE"].ToString())) -
((Int٦٤.Parse(DS.Tables[٠].Rows[i]["TAKHFIF"].ToString()) *
Int٦٤.Parse(DS.Tables[٠].Rows[i]["_COUNT"].ToString()) *
Int٦٤.Parse(DS.Tables[٠].Rows[i]["PRICE"].ToString())) / ١٠٠));
}
textBox_WithoutTakhfif.Text = Sum.ToString();
textBox_WithTakhfif.Text = (Takhfif).ToString();
textBox_Sood.Text = Sood.ToString();
}
```

کد مربوطه به گزارش گیری که تابع report را فراخوانی می کند:

```
private void Form_Reports_Load(object sender, EventArgs e)
{

}

private void button_Applay_Click(object sender, EventArgs e)
{
Report(textBox_FromDate.Text, textBox_ToDate.Text);
}

private void button_Back_Click(object sender, EventArgs e)
{
this.Close();
}


   }
```

کد مربوط به قسمت کاربران در برنامه ی مدیر:

```
namespace BookShop
{
public partial class Form_Users : Form
{
char[] permission = {'0','0','0','0'};
string TypeIndex, TypeName;
IRA ira = new IRA();
DataSet DS = new DataSet();

public Form_Users()
{
InitializeComponent();
}

private void bindList()
{
ira.Select_Record("ACCOUNT", "ID, TYPENAME, NAME, TAKHFIF", out DS, "", "", "", "");
dataGridView_UserList.DataSource = DS;
dataGridView_UserList.DataMember = DS.Tables[0].ToString();

}
```

در ابتدا یک آرایه از نوع کاراکتر برای مجوز های دسترسی تعریف می کنیم.سپس کلاس ira را فراخوانی کرده و یک dataset تعریف می کنیم.

در تابع bind از جدول account نام و شماره و تخفیف و ... را در ds قرار می دهیم وdataset را در یک datagridview میگذاریم.

```
private void button_Back_Click(object sender, EventArgs e)
{
this.Close();
}
```

این دستور برای بستن پنجره ی فعال وبرگشت به پنجره ی قبل می باشد.

کد مربوط به تعریف مدیر:

```
private void
radioButton_Admin_CheckedChanged(object sender, EventArgs e)
{
groupBox_Permission.Enabled = false;
groupBox_Takhfif.Enabled = false;
TypeName = radioButton_Admin.Text;
TypeIndex = radioButton_Admin.Tag.ToString();
}
```

این قسمت کد نشاندهنده ی آن است که اگر کاربری که قرار است تعریف شود یک مدیر است از تخفیف و مجوز های دسترسی که به پرسنل داده می شوند بی نصیب خواهد ماند

کد مربوط به تعریف مشتری:

```
private void radioButton_Customer_CheckedChanged(object sender, EventArgs e)
{
groupBox_Permission.Enabled = false;
groupBox_Takhfif.Enabled = true;
TypeName = radioButton_Customer.Text;
TypeIndex = radioButton_Customer.Tag.ToString();
}
```

اگر کاربری که تعریف می شود از نوع مشتری باشد می تواند تخفیف بگیرد اما دارای مجوز دسترسی نخواهد بود.

کد مربوط به تعریف پرسنل:

```
private void radioButton_Personel_CheckedChanged(object sender, EventArgs e)
{
```

```
groupBox_Permission.Enabled = true;
groupBox_Takhfif.Enabled = false;
TypeIndex = radioButton_Personel.Tag.ToString();
TypeName = radioButton_Personel.Text;
}
```

اگر کاربری که در حال تعریف است جزو پرسنل فروشگاه باشد از تخفیف بی بهره می باشد اما می توان به آن مجوز دسترسی داد.این مجوز ها فقط مخصوص پرسنل فروشگاه می باشد.

```
private void checkBox_Admin_CheckedChanged(object sender, EventArgs e)
{
if ((sender as CheckBox).Checked)
permission[Int32.Parse((sender as CheckBox).Tag.ToString())] = '1';
else
permission[Int32.Parse((sender as CheckBox).Tag.ToString())] = '0';
}
```

این قسمت مربوط به مجوزهای دسترسی می باشد که اگر یک chechbox علامت دار بود بـه معنـای آن است که مجوز دسترسی به ان قسمت را دارد و در غیر این صورت نشان دهنـده ی آن اسـت کـه بـه آن قسمت مجوز دسترسی ندارد.

کد مربوط به قسمت راهنما:

```
private void button_Help_Click(object sender, EventArgs e)
{
string help="1- مجوز دسترسی فقط برای پرسنل است."+ "\n";
help += "2- تخفیف فقط برای مشتری است." + "\n";
help += "3- درصورتی که شناسه تکراری باشد اطلاعات جایگزین می شود." + "\n";
help += "4- سعی شود شناسه از ترکیبی رمز اعداد و حروف باشد." + "\n";
help += "_____" + "\n";
help += "مدیر کاربر بدون تواند نمی سیستم باشد." + "\n";
MessageBox.Show(help, "مدیریت", MessageBoxButtons.OK, MessageBoxIcon.Information,
MessageBoxDefaultButton.Button1, MessageBoxOptions.RtlReading);
}
```

کد تابع برای ثبت اطلاعات:

```
private void button_Save_Click(object sender, EventArgs e)
{
label_Error.Hide();
if (textBox_ID.Text == string.Empty || textBox_Name.Text == string.Empty || textBox_Password.Text ==
string.Empty)
{
label_Error.Show();
return;
}
int permit = 0;
if (radioButton_Admin.Checked) permit = 15;
if (radioButton_Customer.Checked) permit = 0;
if (radioButton_Personel.Checked) ira.PermitEncode(permission, out permit);
ira.Select_Record("ACCOUNT", "ID", out DS, "ID", "=", "OR", textBox_ID.Text);
if (DS.Tables[0].Rows.Count > 0)
ira.Update_Record("ACCOUNT", "ID='" + textBox_ID.Text + "'", "NAME, TYPEINDEX, TYPENAME,
PASSWORD, PERMISSION, TAKHFIF", textBox_Name.Text, TypeIndex, TypeName,
textBox_Password.Text, permit.ToString(), textBox_Takhfif.Text);
else
ira.Insert_Record("ACCOUNT", "ID, NAME, TYPEINDEX, TYPENAME, PASSWORD, PERMISSION,
TAKHFIF", textBox_ID.Text, textBox_Name.Text, TypeIndex, TypeName, textBox_Password.Text,
permit.ToString(), textBox_Takhfif.Text);
bindList();
}
```

در این تابع ابتدا چک می شود که نام و شماره کالا و شماره ی کاربری تهی نباشد که در این صورت پیغام خطا می دهدو از برنامه خارج می شود.

سپس می گوید که اگر در قسمت انتخاب کاربران مدیر انتخاب شده بود به او اجازه ی دسترسی برای کلیه ی امور داده شود و اگر خریدار انتخاب شده بود به او هیچ گونه اجازه ی دسترسی داده نشود.

اگر قرار به انتخاب پرسنل بود بررسی شود که چه مجوز هایی به او داده شده است و مجوز های دسترسی در جدول account ذخیره شود.

سپس بررسی می کند که کالایی که هم اکنون به ثبت رسیده از جمله کالاهایی می باشد کـه از قبـل در فروشگاه وجود داشته و یا اولین بار است که وارد فروشگاه می شود.
اگر کالا را قبلا در فروشگاه داشتیم کافیست که رکورد مربوط به آن کالا را update کنیم و اگـر از قبـل وجود نداشه است باید کالای جدید را در جدول درج کنیم.

تابع مربوط به load شدن مدیریت کاربران:

```
private void Form_Users_Load(object sender, EventArgs e)
{
bindList();
}
```

تابع مربوط به حذف کاربران:

```
private void Form_Users_FormClosing(object sender, FormClosingEventArgs e)
{
ira.Select_Record("ACCOUNT", "ID", out DS, "TYPEINDEX", "=", "AND", "01");
if (DS.Tables[0].Rows.Count < 1)
{
MessageBox.Show("باشد داشته وجود مدیر یک باید حداقل.", "مدیریت", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation,MessageBoxDefaultButton.Button1 , MessageBoxOptions.RtlReading);
e.Cancel = true;
}
else
e.Cancel = false;

}
```

این قسمت از برنامه یرای حذف کردن کاربرانی است که توسط مدیر تعریف شده است.اگر این کاربر از نوع مدیر باشد یعنی شماره ی آن ۰۱ باشد چک می کند که آیا مدیران دیگری در این فروشـگاه وجـود دارند یا خیر.در صورت منفی بودن جواب یک پیغام ظاهر می شود که در آن تذکر می دهد که حـداقل باید یک مدیر وجود داشته باشد.

فرم مربوط به حذف کردن با استفاده از linke حذف شود:

```
private void dataGridView_UserList_CellClick(object sender, DataGridViewCellEventArgs e)
{
if (e.ColumnIndex == 0 && e.RowIndex >=0)
{
if(MessageBox.Show("شود؟ حذف کاربر", "مدیریت", MessageBoxButtons.YesNo, MessageBoxIcon.Question,
MessageBoxDefaultButton.Button2, MessageBoxOptions.RtlReading) == DialogResult.Yes)
ira.Delete_Record("ACCOUNT", "ID", "=", "AND",
dataGridView_UserList.Rows[e.RowIndex].Cells[1].Value.ToString());
bindList();
ira.Select_Record("ACCOUNT", "ID", out DS, "TYPEINDEX", "=", "AND", "01");
if (DS.Tables[0].Rows.Count < 1)
{
string s = "شدند حذف مدیر کاربران تمامی." + "\n";
s += "باشد داشته وجود مدیر کاربر یک باید حداقل." + "\n";
MessageBox.Show(s, "مدیریت", MessageBoxButtons.OK, MessageBoxIcon.Exclamation,
MessageBoxDefaultButton.Button1, MessageBoxOptions.RtlReading);
}
bindList();
}
}
```

این قسمت برنامه مربوط به linke حذف شود در اولین سلول datagridview می باشد.اگر این گزینه انتخاب شود در ابتدا با یک پیغام صحت دستور وارد شده را بررسی می کند و سپس چک می کند که این کاربر یک کاربر مدیر است یا خیر.اگر کاربر مدیریت حذف شود و این مدیر تنها مدیر موجود در فروشگاه باشد پیغام می دهد که فروشگاه مدیر ندارد و حداقل باید دارای یک مدیر باشد.

فرم مربوط به قسمت تخفیف:

```
private void textBox_Takhfif_TextChanged(object sender, EventArgs e)
{
foreach (char chr in (sender as TextBox).Text)
if (chr < '0' || chr > '9')
{
(sender as TextBox).Text = "0";
break;
}
if ((sender as TextBox).Text == string.Empty) (sender as TextBox).Text = "0";
}

private void textBox_Takhfif_KeyPress(object sender, KeyPressEventArgs e)
{
if ((e.KeyChar < '0' || e.KeyChar > '9') && e.KeyChar != '\b')
e.KeyChar = '\0';
}
}
```

همانطور که قبلا گفته شد فقط خریدارن می توانند از تخفیف در فروشگاه استفاده کنند.این قسمت از کد برنامه برای این است که قسمت مربوط به تخفیف محدود شود به استفاده از عدد و دکمه ی backspace .همچنین کاربران نمی توانند با استفاده از کپی رشته ایی را در این قسمت وارد کنند.

کد مربوط به انتخاب عکس در برنامه ی مدیر:

```
public partial class SelectPicture : UserControl
{
bool _haspic = false;
public SelectPicture()
{
InitializeComponent();
}

private void openFileDialog1_FileOk(object sender, CancelEventArgs e)
{
pictureBox_IMG.Load(openFileDialog1.FileName);
_haspic = true;
}

public bool HasPic()
{
return _haspic;
}


private void openToolStripButton_Click(object sender, EventArgs e)
{
openFileDialog1.ShowDialog();
}

private void newToolStripButton_Click(object sender, EventArgs e)
{
pictureBox_IMG.ImageLocation = null;
pictureBox_IMG.Image = null;
pictureBox_IMG.Refresh();
_haspic = false;
}
}
}
```

در این قسمت برنامه ابتدا یک متغیر بولی تعریف کردیم به نام haspic_ و مقدار اولیه ی آن را false در نظر می گیریم.یعنی فرض می کنیم که هیچ عکسی در ابتدا وجود ندارد.سپس در هنگام load عکس این متغیر مقدار true می گیرد.

غرفه ی لوازم التحریر در برنامه ی مدیر:

کد تابع مربوط به bind:

```
private void bind()
{
ira.Select_Record("TYPEWRITE", "ID, NAME, PRICE, _COUNT", out DS, "GID", "=", "AND",
comboBox_Group.SelectedValue.ToString());
dataGridView_TypeWriteList.DataSource = DS;
dataGridView_TypeWriteList.DataMember = DS.Tables[0].ToString();
}
```

در این قسمت از برنامه خواسته شده است که رکورد هایی را از جدول لوازم التحریر نشان دهد که شماره ی گروه آن با چیزی که مد نظر است هم خوانی داشته باشد. این اطلاعات در یک Ds ریخته می شود و سپس این ds در datagridview خواسته شده نمایش داده می شود.

تابع جستجو در غرفه ی لوازم التحریر:

```
private void Filter(string _Filed, string _Operator, string _Value)
{
SqlConnection Connection = new SqlConnection("Data Source=localhost;Initial Catalog=Shop;Integrated Security=True");
SqlCommand Command = new SqlCommand();
SqlDataAdapter Adapter = new SqlDataAdapter();
DS.Clear();
Command.CommandText = "SELECT ID, NAME, PRICE, _COUNT FROM TYPEWRITE WHERE GID='" + comboBox_Group.SelectedValue.ToString() +"' AND " + _Filed + _Operator + _Value;
Command.Connection = Connection;
Connection.Open();
Adapter.SelectCommand = Command;
Adapter.Fill(DS);
Connection.Close();
}
```

در ابتدای تابع فیلتر یک connection تعریف کرده ایم که با توجه به آن میبینیم که "datasource=localhost" قرار داده ایم و این به این خاطر می باشد که می خواهیم این کار ها بر روی سیستم فعلی یعنی سیستمی که هم اکنون در حال استفاده می باشد انجام شود و اگر فروشگاه به شبکه وصل شود با توجه به مشگلات امنیتی و اینکه هر کاربری با دستکاری کردن این داده مشگل ناسازگاری به وجود نیاورد استفاده شده است.

در خط بعدی یک command تعریف کرده ایم برای اجرای دستورات sql و در دستور بعدی یک adpter تعریف شده برای خواندن رکوردها از بانک اطلاعاتی.

در دستور بعدی ds را پاک میکنیم تا اگر از اطلاعات قبلی در آن موجود باشد بر روند کار اثری نداشته باشد .

تابع فیلتر برای جستجو کردت طراحی شده است و به کمک combobox هایی که در سمت راست طراحی شده است این کار انجام می پذیرد.

در زمانی که comnobox های مربط به جستجو تغییر می کنند این نابع فراخوانی می شود.

تابع مربوط به بستن پنجره ی فعال:

```
private void button_Back_Click(object sender, EventArgs e)
{
this.Close();
}
```

تابع مربوط به پاک کردن اطلاعات:

```
private void button_Clear_Click(object sender, EventArgs e)
{
label_Error.Hide();
textBox_ID.Clear();
textBox_Name.Clear();
textBox_Note.Clear();
textBox_Price.Clear();
textBox_Count.Clear();
textBox_ID.Focus();
}
```

ابتدا پیغام خطای مربوط به اشتباه در ورود اطلاعات پاک شده و سپس textbox های مربوط بـه شـماره ، نام ، توضیحات ، قیمت و تعداد پاک می شود و مکان نما در اولین سطر یعنی Textbox مربوط به Id قرار می گیرد.

کد تابع مربوط به ثبت اطلاعات:

```
private void button_Save_Click(object sender, EventArgs e)
{
label_Error.Hide();
if (textBox_ID.Text == string.Empty || textBox_Price.Text == string.Empty || textBox_Count.Text ==
string.Empty)
{
label_Error.Show();
return;
}
```

در دستورات فوق ابتدا چک شده است که اگر شماره ی کالا ، نام ، قیمت یا تعداد وارد نشده است به کالای مورد نظر ترتیب اثر داده نشود و از برنامه خارج شود.

کد اتصال رشته به database:

```
//-- بیس دیتا به اتصال رشته
SqlConnection Connection = new SqlConnection("Data Source=localhost;Initial Catalog=Shop;Integrated Security=True");
SqlCommand Command = new SqlCommand();
//-- بافر در بایت صورت به تصویر قراردادن
byte[] buff = new byte[1];
if (selectPicture1.HasPic())
{
System.IO.StreamReader pic = new
System.IO.StreamReader(selectPicture1.pictureBox_IMG.ImageLocation);
long len = pic.BaseStream.Length;
buff = new byte[pic.BaseStream.Length];
pic.BaseStream.Read(buff, 0, Convert.ToInt32(len));
}
```

در کد فوق در ابتدا رشته مورد نظر به database متصل شده است و در دستور بعد یک آرایه از نوع بایت تعریف شده است.

در دستور بعد چک شده است که آیا تصویری وجود دارد یا خیر.در صورت درست بودن عبارت فوق این تصویر در ram ذخیره می شود.

سپس یک متغیر به نام len تعریف شده است که طول تصویر را در خود ذخیره می کند.

سپس تصویر مورد نظر در متغیر buff که در بالا تعریف شد قرار گرفته و سپس از ram خوانده می شود.

کد تابع برای بررسی وجود رکورد:

```
//----------- وجود برای بررسی
int count = 0;
ira.Select_Record("TYPEWRITE", "_COUNT", out DS, "GID, ID", "=", "AND",
comboBox_Group.SelectedValue.ToString(), textBox_ID.Text.Trim());
if (DS.Tables[0].Rows.Count <= 0)
{
```

در این قسمت برنامه گفته شده که رکورد هایی را نمایش بده که که شماره ی گروه و شماره ی کالا با چیزی که مدنظر ماست یکی باشد.

سپس بررسی شده است که آیا چنین رکوردی وجود دارد یا نه؟

اگر وجود نداشته باشد پس باید آن را ذر بانک اطلاعای خود درج کنیم.

کد مربوط به درج رکورد:

```
// --  دیتابیس در رکورد درج
Command.CommandText = "INSERT INTO TYPEWRITE(GID, ID, NAME, PRICE, _COUNT, NOTE,
PIC) VALUES(@GID, @ID, @NAME, @PRICE, @COUNT, @NOTE, @PIC)";
}
else
{
// --  موجود رکورد رسانی روز به
count = Int32.Parse(DS.Tables[0].Rows[0]["_COUNT"].ToString());
Command.CommandText = "UPDATE TYPEWRITE SET  NAME=@NAME, PRICE=@PRICE,
_COUNT=@COUNT, NOTE=@NOTE, PIC=@PIC WHERE GID=@GID AND ID=@ID";
}
```

اگر رکورد مورد نظر قبلا در بانک وجود داشت عملیات به روز رسانی و در غیر این صورت عملیـات درج انجام می شود.

```
//--- دیتابیس در مقادیر درج برای پارامتر افزودن
Command.Parameters.Clear();
Command.Parameters.Add("@GID", SqlDbType.NVarChar);
Command.Parameters.Add("@ID", SqlDbType.NVarChar);
Command.Parameters.Add("@NAME", SqlDbType.NVarChar);
Command.Parameters.Add("@PRICE", SqlDbType.NVarChar);
Command.Parameters.Add("@COUNT", SqlDbType.NVarChar);
Command.Parameters.Add("@NOTE", SqlDbType.NText);
Command.Parameters.Add("@PIC", SqlDbType.Image);
```

کد مربوط به بارگذاری فرم لوازم التحریر:

```
private void Form_TypeWrite_Load(object sender, EventArgs e)
{
// TODO: This line of code loads data into the 'shopDataSet_Group._GROUP' table. You can move, or
remove it, as needed.
this._GROUPTableAdapter.FillBy (this.shopDataSet_Group._GROUP, "04");
bind();
comboBox_IDFilter.SelectedIndex = 0;
comboBox_NameFilter.SelectedIndex = 0;
comboBox_PriceFilter.SelectedIndex = 0;
comboBox_CountFilter.SelectedIndex = 0;
}
```

در ابتدا برای لیست کردن گروه ها combobox‌های گروه را به بانک اطلاعاتی اتصال می دهیم و سپس حالت پیش فرض تمامی گروه ها را شماره ی صفر یعنی اولین گزینه قرار می دهیم.

در دستور زیر گفته شده که هر بار گزینه هایی موجود در combobox ها تغییر کرد ابع bind فراخوانی شود.

```
private void comboBox_Group_SelectedIndexChanged(object sender, EventArgs e)
{
bind();
}
```

کد مربوط به linke حذف شود:

```
private void dataGridView_TypeWriteList_CellClick(object sender, DataGridViewCellEventArgs e)
{
if (e.ColumnIndex == 0 && e.RowIndex >= 0) //-- حذف ی دکمه کد
{
string msg = dataGridView_TypeWriteList.Rows[e.RowIndex].Cells[3].Value.ToString() + " شود؟ حذف ";
if (MessageBox.Show(msg, "مدیرت", MessageBoxButtons.YesNo, MessageBoxIcon.Question,
MessageBoxDefaultButton.Button1, MessageBoxOptions.RtlReading) == DialogResult.Yes)
{
ira.Delete_Record("TYPEWRITE", "GID, ID", "=", "AND", comboBox_Group.SelectedValue.ToString(),
dataGridView_TypeWriteList.Rows[e.RowIndex].Cells[2].Value.ToString());
bind();
}
```

کد فوق مربوط به اولین سلول موجود در datagridview به نام "حذف شود؟" می باشد.
اگر این گزینه انتخاب شود پیغامی برای تایید عملیات انجام شده داده می شود و پس از تایید پیغام از جدول لوازم التحریر رکورد هایی حذف می شود که شماره ی گروه و شماره ی کالای آن با کالایی که انتخاب شده یکی باشد.

```
//-------- آن نمایش و دیتابیس از عکس خواندن
Command.Connection = Connection;
```

```
Command.Parameters.Clear();
Command.Parameters.Add("@GID", SqlDbType.NVarChar);
Command.Parameters.Add("@ID", SqlDbType.NVarChar);
Command.Parameters["@GID"].Value = comboBox_Group.SelectedValue.ToString();
Command.Parameters["@ID"].Value =
dataGridView_TypeWriteList.Rows[e.RowIndex].Cells[2].Value.ToString().Trim();
Command.CommandText = "SELECT PIC FROM TYPEWRITE WHERE GID=@GID AND ID=@ID";
Connection.Open();
byte[] pic = (byte[])Command.ExecuteScalar();
Connection.Close();
if (pic.Length > 1)
{
System.IO.MemoryStream stream = new System.IO.MemoryStream();
stream.Write(pic, 0, pic.Length);
Bitmap img = new Bitmap(stream);
selectPicture1.pictureBox_IMG.Image = img;
}
else
selectPicture1.pictureBox_IMG.Image = null;

}
}
```

بعد از باز کردن connection و پاک کردن پارامتر های قبلی آن اطلاعات جدید مربوط به نام ،شماره ی کالا ، شماره گروه به آن اضافه می شود.

سپس در قالب یک دستور sql خواسته شده که از جدول لوازم التحریر آنهایی را که شماره ی گروه و شماره ی کالای آن با چیزی که مد نظر ماست یکی باشد را انتخاب کن.

بعد از باز کردن Connection یک آرایه با نام pic تعریف شده است .اگر عکس با مشخصات فوق پیدا شد آن را در memory stream برده و سپس memorystream را به عنوان ورودی به یک Bimap داده و سپس آن را به نمایش در می آورد.

فرم مربوط به وارد کردن قیمت:

```
private void textBox_Price_TextChanged(object sender, EventArgs e)
{
foreach(char chr in (sender as TextBox).Text)
if (chr < '0' || chr > '9')
{
(sender as TextBox).Text = "0";
return;
}
if ((sender as TextBox).Text == string.Empty) (sender as TextBox).Text = "0";
}

private void textBox_Price_KeyPress(object sender, KeyPressEventArgs e)
{
if ((e.KeyChar < '0' || e.KeyChar > '9') && e.KeyChar != '\b')
e.KeyChar = '\0';
}
```

در برنامه ی فوق texbox های مربوط به قیمت محدود به گرفتن اعداد ۰ تا ۹ و کلید backspace می باشد.

همچنین کاربر برای ورود رشته قادر به انجام عملیات کپی نیز نمی باشد.

کد تابع مربوط به جستجو از طریق شماره:

```
private void comboBox_IDFilter_SelectedIndexChanged(object sender, EventArgs e)
{
if (textBox_ID.Text == string.Empty) bind();
switch (comboBox_IDFilter.SelectedIndex)
{
case 0: bind(); break;
case 1: Filter("ID", " LIKE '", textBox_ID.Text + "'"); break;
case 2: Filter("ID", "='", textBox_ID.Text + "'"); break;
}
}
```

در این قسمت برنامه تابع فیلتر فراخوانی شده است به این ترتیب که در ابتدا چک می شود که کدام گزینه برای جستجو انتخاب شده است.اگر انتخاب بر روی index شماره ی صفر باشد یعنی گزینه ی همه انتخاب شده است و در این صورت تابع bind فراخوانی می شود.

در غیر این صورت اگر گزینه ی شبیه انتخاب شود تمام کالاهایی که شبیه بر کالای مورد نظر بوده لیست می شود و اگر گزینه ی برابر انتخاب شود تمام کالاهایی که دقیقا برابر با کالای مورد نظر باشد لیست می شوند.

کد تابع جستجو بر اسا س نام:

```
private void comboBox_NameFilter_SelectedIndexChanged(object sender, EventArgs e)
{
if (textBox_Name.Text == string.Empty) bind();
switch (comboBox_NameFilter.SelectedIndex)
{
case 0: bind(); break;
```

```
case 1: Filter("NAME", " LIKE '", textBox_Name.Text + "'"); break;
case 2: Filter("NAME", "='", textBox_Name.Text + "'"); break;
}
}
```

در این قسمت برنامه تابع فیلتر فراخوانی شده است به این ترتیب که در ابتدا چک می شود که کدام گزینه برای جستجو انتخاب شده است.اگر انتخاب بر روی index شماره ی صفر باشد یعنی گزینه ی همه انتخاب شده است و در این صورت تابع bind فراخوانی می شود.

در غیر این صورت اگر گزینه ی شبیه انتخاب شود تمام کالاهایی که در نام شبیه بر کالای مورد نظر بوده لیست می شود و اگر گزینه ی برابر انتخاب شود تمام کالاهایی که در نام دقیقا برابر با کالای مورد نظر باشد لیست می شوند.

فرم مربوط به برنامه ی راهنما:

```
private void button_Help_Click(object sender, EventArgs e)
{
Form_TypeWriteHelp Help = new Form_TypeWriteHelp();
Help.ShowDialog();
}
}
```

فرم مربوط به پوستر در برنامه ی خریدار:

```
public partial class Form_Pooster : Form
{
string BascketID = "";
string Sood = "0";
IRA ira = new IRA();
SqlConnection Connection = new SqlConnection("Data Source=localhost;Initial Catalog=Shop;Integrated
Security=True");
SqlCommand Command = new SqlCommand();
SqlDataReader Reader;
DataSet DS=new DataSet();
int ImgIndex = 0;
```

در ابتدا یک متغیر رشته ایی به نام "bascket id" با مقدار تهی تعریف می کنیم که شماره ی سبد را در آن قرار دهیم.

سپس یک متغیر رشته ایی برای قرار دادن مقدار سود تعریف می کنیم.

برای استفاده از کلاس ira آن را در برنامه ی موجود فراخوانی می کنیم.

سپس یک "connection" تعریف می کنیم.برای اجرای دستورات از "sqlcommand" و برای خواندن رکورد ها از یک "datareader" استفاده می کنیم.

برای قرار دادن این مقادیر نیز یک dataset تعریف می کنیم.

کد تابع bind:

```
//--------------------

private void bind()
{
ira.Select_Record("POOSTER", "ID, NAME, PRICE, NOTE, _COUNT", out DS, "GID, TYPE", "=",
"AND", comboBox_Group.SelectedValue.ToString(), comboBox_Type.SelectedIndex.ToString());
if (DS.Tables[0].Rows.Count > 0)
{
ImgIndex = 0;
textBox_ID.Text = DS.Tables[0].Rows[ImgIndex]["ID"].ToString();
textBox_Title.Text = DS.Tables[0].Rows[ImgIndex]["NAME"].ToString();
textBox_Price.Text = DS.Tables[0].Rows[ImgIndex]["PRICE"].ToString();
textBox_Note.Text = DS.Tables[0].Rows[ImgIndex]["NOTE"].ToString();
textBox_CurrentCount.Text = DS.Tables[0].Rows[ImgIndex]["_COUNT"].ToString();
ShowPic(0);
}
else
{
textBox_ID.Text = "";
textBox_Title.Text = "";
textBox_Price.Text = "";
textBox_Note.Text = "";
textBox_CurrentCount.Text = "";
pictureBox_Image.Image = null;
}
}
```

سپس در تابع bind با دستور Sql اول می گوییم که از جدول پوستر شماره و نـام و قیمـت و تعـداد و... آنهایی را نمایش بده که شماره ی گروه و نوع آن با چیزی که ما وارد کردیم یکی باشد.

سپس گفته شده اگر چیزی با این مشخصاتی که گفته شد وجود دارد مشخصات گفته شـده را در textbox های گفته شده نمایش بده و اگرچیزی با این مشخصات یافت نشد textbox های مربوطه را با مقدار خالی پر کن.

تابع بعدی برای نمایش عکس می باشد:

```
private void ShowPic(int _Index)
{
textBox_ID.Text = DS.Tables[0].Rows[ImgIndex]["ID"].ToString();
textBox_Title.Text = DS.Tables[0].Rows[ImgIndex]["NAME"].ToString();
textBox_Price.Text = DS.Tables[0].Rows[ImgIndex]["PRICE"].ToString();
textBox_Note.Text = DS.Tables[0].Rows[ImgIndex]["NOTE"].ToString();
textBox_CurrentCount.Text = DS.Tables[0].Rows[ImgIndex]["_COUNT"].ToString();
System.IO.MemoryStream Stream = new System.IO.MemoryStream();
Command.Connection = Connection;
Command.CommandText = "SELECT PIC FROM POOSTER WHERE GID='" +
comboBox_Group.SelectedValue.ToString() + "' AND TYPE='" +
comboBox_Type.SelectedIndex.ToString() + "' AND ID='" + textBox_ID.Text.Trim() + "'";
Connection.Open();
byte[] image = (byte[])Command.ExecuteScalar();
Connection.Close();
if (image != null)
{
if (image.Length > 1)
{
Stream.Write(image, 0, image.Length);
Bitmap bitmap = new Bitmap(Stream);
pictureBox_Image.Image = bitmap;
}
else
pictureBox_Image.Image = null;
}
}
```

در ابتدا مشخصات عکس را وارد textbox های مربوطه کردیم. سپس یک "memorystream" تعریف کردیم که عکسها را داخل ram نگهداری کند.سپس برای استفاده از پایگاه داده connection را باز کرده و در یک دستور Sql مشخصات عکس هایی را خواستیم از جدول پوستر که شماره ی گروه و نوع عکس ها با مشخصاتی که ما می خواهیم همخوانی داشته باشد.

سپس یک آرایه از نوعbyte تعریف می کنیم که مشخصات عکس را به طور ترتیبی بخواند و byte که در سمت راست این تعریف آرایه تکرار شده است برای تبدیل نوع می باشد و بعد از اتمام کار connection را می بندیم.

سپس بررسی می کنیم که اگر عکسی با مشخصات فوق الذکر وجود داشته است آن عکس داخل "stream' نوشته شود.

سپس یک آرایه از نوع bitmap تعریف شده است ودلیل تعریف این نوع آرایه این است که عکس ها به صورت باینری ذخیره شده اند و به همین دلیل به bitmap نیازمندیم.منبعی که برای Bitmap در نظر گرفته شده است "Stream" می باشد یعنی عکس هایی که در "stream" هستند توسط Bitmap نشان داده می شوند و اگر عکسی برای نمایش دادن یافت نشد مقدار null در picturebox قرار می گیرد.

```
public Form_Pooster(string _BascketID)
{
InitializeComponent();
BascketID = _BascketID;
}
```

پارامتر bascketid_ به صورت متغیر محلی می باشد و چون می خواهیم در برنامه از آن استفاده کنیم آن را در داخل یک متغیر سراسری می ریزیم.

این تابع برای نمایش سود و شماره ی سبد می باشد:

```
private void Form_Pooster_Load(object sender, EventArgs e)
{
ira.Select_Record("SOOD", "POOSTER", out DS, "", "", "", "");
if (DS.Tables[0].Rows.Count > 0)
Sood = DS.Tables[0].Rows[0]["POOSTER"].ToString();
DS.Reset();
// TODO: This line of code loads data into the 'shopDataSet._GROUP' table. You can move, or remove it, as needed.
this._GROUPTableAdapter.FillBy(this.shopDataSet._GROUP, "03");
```

```
Text += "                                                    شما سبد شماره:" + BascketID;
comboBox_Type.SelectedIndex = 0;
bind();
}
```

این تابع داده ها را از جدول shopdataset می خواند اگر عکس وجود داشت سود آن را محاسبه کرده و همراه با بقیه ی اطلاعات و شماره ی سبد مشتری به وی نشان می دهد.
برای اتصال به بانک ازدستور fillby استفاده شده و در انتها تابع bind فراخوانی شده است.

این تابع برای نمایش تعداد عکس ها می باشد:

```
private void textBox_Count_KeyPress(object sender, KeyPressEventArgs e)
{
if ((e.KeyChar < '0' || e.KeyChar > '9') && e.KeyChar != '\b')
e.KeyChar = '\0';
}

private void textBox_Count_TextChanged(object sender, EventArgs e)
{
foreach (char chr in (sender as TextBox).Text)
if (chr < '0' || chr > '9')
{
(sender as TextBox).Text = "0";
return;
}
if ((sender as TextBox).Text == string.Empty) (sender as TextBox).Text = "0";
}
```

در این قسمت texbox مربوط به تعداد محدود به گرفتن عدد و backspace می باشد تا کاربر بتواند اطلاعات غلط خود را پاک کند.
همچنین کاربر نمی تواند از عملیات کپی برای گنجاندن رشته در این قسمت استفاده کند و اگر کاربر اطلاعات درست در این قسمت وارد نکند texbox مربوطه مقدار پیش فرض صفر به آن تعلق می گیرد.

این تابع برای نمایش سبد خرید می باشد:

```
private void linkLabel_Bascket_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
```

```
{
Form_Bascket Bascket = new Form_Bascket(BascketID);
Bascket.ShowDialog();
}
```

این تابع برای بستن پنجره ی فعال می باشد:

```
private void linkLabel_Back_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
this.Close();
}
```

این قسمت برای نمایش تصویر در اندازه ی واقعی می باشد:

```
private void linkLabel_ViewImage_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
Form_ViewImage ViewImage = new Form_ViewImage(pictureBox_Image.Image);
ViewImage.ShowDialog();
}
```

این قسمت برای نمایش اطلاعات با انتخاب کاربر می باشد.کاربر می تواند پوستر ،نقشه و یا کارت پستار را در این قسمت انتخاب کند:

```
private void comboBox_Type_SelectedIndexChanged(object sender, EventArgs e)
{
bind();
}
```

بعد از انتخاب هر گروه برای نمایش تابع Bind فراخوانی می شود تا اطلاعات لازم نمایش داده شوند.

این قسمت برای تغییر متن می باشد:

```
private void textBox_ID_TextChanged(object sender, EventArgs e)
{
foreach (char chr in (sender as TextBox).Text)
if (chr < '0' || chr > '9') (sender as TextBox).Clear();
if ((sender as TextBox).Text.Length > 0)
linkLabel_GotoBascket.Enabled = true;
else
linkLabel_GotoBascket.Enabled = false;
}
```

در قسمت برو تو سبد چک می کند که این رشته ی وارد شده عدد باشد و همچنین بزرگتر از صفر باشد که در این صورت تقاضا وارد سبد خرید می شود.

این قسمت برای نمایش عکس بعدی می باشد:

```
private void linkLabel_NextImage_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
if(DS.Tables[0].Rows.Count > 0)
if (ImgIndex < DS.Tables[0].Rows.Count - 1)
ShowPic(++ImgIndex);
else
{
ImgIndex = 0;
ShowPic(ImgIndex);
}
}
```

اگر تعداد عکس های موجود بیش از یکی باشد آنگاه عکس بعدی را نمایش می دهد و در غیر این صورت باز به اولین عکس رجوع می شود.

این قسمت برای نمایش عکس قبلی استفاده می شود:

```
private void linkLabel_PrevioseImage_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
if (DS.Tables[0].Rows.Count > 0)
if (ImgIndex > 0)
ShowPic(--ImgIndex);
else
{
ImgIndex = DS.Tables[0].Rows.Count - 1;
ShowPic(ImgIndex);
```

```
}
}
```

اگر تعداد عکس ها بیشتر از یکی بود و عکسی که الان در حال نمایش است اولین عکس موجود در پایگـاه داده نبود آنگاه عکس قبلی را نمایش بده.

کد مربوط به link برو تو سبد:

```
private void linkLabel_GotoBascket_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
if (textBox_ID.Text == string.Empty) return;
label_Error.Hide();
if (Int32.Parse(textBox_Count.Text) < 1)
{
label_Error.Show();
return;
}
if (Int32.Parse(textBox_Count.Text) > Int32.Parse(textBox_CurrentCount.Text))
{
MessageBox.Show(".است موجود تعداد از بیشتر شده وارد تعداد", "فروشگاه", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
textBox_Count.SelectAll();
textBox_Count.Focus();
return;
}
```

در این قسمت در ابتدا بررسی شده است که عکس با شماره ی موجود وجود داشته باشد و شماره ی عکس برابر با یک رشته ی تهی نباشد.

در این هنگام بررسی می شود که تعداد وارد شده بیشتر از تعداد موجود نباشد که در این حالت پیغام می دهد که تعداد وارد شده بیشتر از تعداد موجود است.

کد تابع update:

```
//--- افزوده قبلی تعداد به جدید تعداد حالت این در که قبلی درج برای بررسی
int Count = 0;
Command.CommandText = "SELECT _COUNT FROM BASCKET WHERE BASCKETID='" + BascketID
+ "' AND ID='03" + comboBox_Group.SelectedValue.ToString() + textBox_ID.Text + "'";
Command.Connection = Connection;
Connection.Open();
Reader = Command.ExecuteReader();
if (Reader.Read()) //--- شود رسانی روز به تعداد باید و شده درج قبلا رکورد که حالتی
{
//----- رکورد رسانی روز به و قبلی تعداد به تعداد افزودن
Count = Int32.Parse(Reader[0].ToString()) + Int32.Parse(textBox_Count.Text);
ira.Update_Record("BASCKET", "BASCKETID='" + BascketID + "' AND ID='03" +
comboBox_Group.SelectedValue.ToString() + textBox_ID.Text + "'", "_COUNT", Count.ToString());
}
```

کد تابع درج رکورد:

```
else //--- است نشده درج قبلاً کالا که حالتی
ira.Insert_Record("BASCKET", "BASCKETID, ID, _COUNT, NAME, PRICE, SOOD, SHOPID, GID,
MID", BascketID, "03"+ comboBox_Group.SelectedValue.ToString()+ textBox_ID.Text,
textBox_Count.Text, comboBox_Type.Text + " " + textBox_Title.Text, textBox_Price.Text, Sood, "03",
comboBox_Group.SelectedValue.ToString(), textBox_ID.Text);
Connection.Close();
//---- سبد در کالا درج پیام نمایش
if (checkBox_GotoBascket.Checked)
{
string msg = "تعداد " + textBox_Count.Text + comboBox_Type.Text + "  " + textBox_Title.Text + " سبد در
شد داده قرار. ";
MessageBox.Show(msg, "مدیریت", MessageBoxButtons.OK, MessageBoxIcon.Information,
MessageBoxDefaultButton.Button1, MessageBoxOptions.RtlReading);
}
}
```

این قسمت برنامه در ابتدا کالای مورد نظر را با اطلاعاتش نشان میدهد .اگر این کالا قبلا در سبد خرید درج شده باشد و اکنون تعداد دیگری به آن اضافه شده باشد در آن هنگام تعداد با تعداد قبلی جمع شده و به روز رسانی می شود.

اگر قبلا درج نشده باشد این تعداد در سبد خرید قرار می گیرد و در انتها با یک پیغام نشان می دهد که چه تعداد از کالای مورد نظر در سبد قرار داده شده است.

قسمت بعد برای نمایش راهنما می باشد:

```
private void linkLabel_Help_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
Form_BuyHelp Help = new Form_BuyHelp();
Help.ShowDialog();
}

}
```

قسمت بعد کد مربوط به قسمت لوازم التحریر در برنامه ی خریدار می باشد:

```
public partial class Form_TypeWrite : Form
{
string BascketID = "";
string Sood = "۰";
IRA ira = new IRA();
SqlConnection Connection = new SqlConnection("Data Source=localhost;Initial Catalog=Shop;Integrated Security=True");
SqlCommand Command = new SqlCommand();
SqlDataReader Reader;
DataSet DS = new DataSet();
```

در ابتدا یک متغیر رشته ایی به نام "bascket id" با مقدار تهی تعریف می کنیم که شماره ی سبد را در آن قرار دهیم.
سپس یک متغیر رشته ایی برای قرار دادن مقدار سود تعریف می کنیم.
برای استفاده از کلاس ira آن را در برنامه ی موجود فراخوانی می کنیم.
سپس یک connection تعریف می کنیم.برای اجرای دستورات از sqlcommand و برای خواندن رکورد ها از یک datareader استفاده می کنیم.
برای قرار دادن این مقادیر نیز یک dataset تعریف می کنیم.
کد تابع bind:

```
//----------------------
private void bind()
{

textBox_Note.Clear();
ira.Select_Record("TYPEWRITE", "ID, NAME, PRICE, NOTE, _COUNT", out DS, "GID", "=", "AND", comboBox_Group.SelectedValue.ToString());
dataGridView_TypeWriteList.DataSource = DS;
dataGridView_TypeWriteList.DataMember = DS.Tables[۰].ToString();
}
```

سپس در تابع bind با دستور Sql اول می گوییم که از جدول لوازم التحریر شماره و نام و قیمت و تعداد و... آنهایی را نمایش بده که شماره ی گروه و نوع آن با چیزی که ما وارد کردیم یکی باشد.
سپس گفته شده اگر چیزی با این مشخصاتی که گفته شد وجود دارد مشخصات گفته شده را در datagridevieوهای گفته شده نمایش بده.

```
public Form_TypeWrite(string _BascketID)
{
```

```
InitializeComponent();
BascketID = _BascketID;
}
```

پارامتر baskcetid_ به صورت متغیر محلی می باشد و چون می خواهیم در برنامه از آن استفاده کنیم آن را در داخل یک متغیر سراسری می ریزیم.

کد مربوط به بارگذاری فرم لوازم التحریر:

```
private void Form_TypeWrite_Load(object sender, EventArgs e)
{
ira.Select_Record("SOOD", "TYPEWRITE", out DS, "", "", "", "");
if (DS.Tables[٠].Rows.Count > ٠)
Sood = DS.Tables[٠].Rows[٠]["TYPEWRITE"].ToString();
DS.Reset();
// TODO: This line of code loads data into the 'shopDataSet._GROUP' table. You can move, or remove it, as needed.
this._GROUPTableAdapter.FillBy (this.shopDataSet._GROUP, "٠٤");
:" + BascketID;شماره سبد شماText += "
bind();
}
```

این تابع داده ها را از جدول "shopdataset"می خواند اگر عکس وجود داشت سود آن را محاسبه کرده و همراه با بقیه ی اطلاعات و شماره ی سبد مشتری به وی نشان می دهد.

کد مربوط به بستن پنجره ی فعال:

```
private void linkLabel_Back_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
this.Close();
}
```

این قسمت برای بستن پنجره ی فعال و برگشتن به فرم قبلی می باشد.

کد مربوط به link سبد خرید:

```
private void linkLabel_Bascket_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
Form_Bascket Bascket = new Form_Bascket(BascketID);
Bascket.ShowDialog();
}
```

هنگام فعال شدن این link فرم مربوط به سبد خرید نمایش داده می شود.

این قسمت برای نمایش تصویر در اندازه ی واقعی می باشد:

```
private void linkLabel_ViewImage_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
    {
      Form_ViewImage ViewImage = new Form_ViewImage(pictureBox_Image.Image);
      ViewImage.ShowDialog();
    }
```

کد تابع مربوط به تغییر نوع کالا:

```
private void comboBox_Group_SelectedIndexChanged(object sender, EventArgs e)
{
bind();
}
```

این قسمت برای تعویض کالای مورد نیاز می باشد به این شکل که اگر مداد انتخاب شد با استفاده از تـابع bind مشخصات مربوط به آن و اگر دفتر انتخاب شد مشخصات مربوط به آن نمایش داده شود.

```
private void dataGridView_TypeWriteList_CellClick(object sender, DataGridViewCellEventArgs e)
{
if (e.RowIndex >= ۰)
{
label_Error.Hide();
کد دکمه توضیحات بیشتر if (e.ColumnIndex == ۰) //----
{
System.IO.MemoryStream Stream = new System.IO.MemoryStream();
//-----------------------
textBox_Note.Text =
dataGridView_TypeWriteList.Rows[e.RowIndex].Cells["Column_Note"].Value.ToString();
Command.Connection = Connection;
Command.CommandText = "SELECT PIC FROM TYPEWRITE WHERE GID='" +
comboBox_Group.SelectedValue.ToString() + "' AND ID='" +
dataGridView_TypeWriteList.Rows[e.RowIndex].Cells["Column_ID"].Value.ToString() + "'";
Connection.Open();
byte[] image = (byte[])Command.ExecuteScalar();
Connection.Close();
if (image.Length > ۱)
{
Stream.Write(image, ۰, image.Length);
Bitmap bitmap = new Bitmap(Stream);
pictureBox_Image.Image = bitmap;
}
else
pictureBox_Image.Image = null;
}
```

در ابتدا مشخصات عکس را وارد textbox های مربوطه کردیم. سپس یک "memorystream" تعریـف کردیم که عکسها را داخل ram نگهداری کند.سپس برای استفاده از پایگاه داده connection را باز کرده و در یک دستور Sql مشخصات عکس هایی را خواستیم از جدول پوستر که شماره ی گروه و نوع عکـس ها با مشخصاتی که ما می خواهیم همخوانی داشته باشد.

سپس یک آرایه از نوعbyte تعریف می کنیم که مشخصات عکس را به طور ترتیبی بخواند و byte کـه در سمت راست این تعریف آرایه تکرار شده است برای تبدیل نوع می باشد و بعد از اتمام کار connection را می بندیم.

سپس بررسی می کنیم که اگر عکسی با مشخصات فوق الذکر وجود داشته است آن عکس داخل stream نوشته شود.

سپس یک آرایه از نوع bitmap تعریف شده است ودلیل تعریف این نوع آرایه این است که عکس ها بـه صورت باینری ذخیره شده اند و به همین دلیل به bitmap نیازمندیم.منبعی کـه بـرای Bitmap در نظـر گرفته شده است Stream می باشد یعنی عکس هایی که در stream هستند توسـط Bitmap نشـان داده می شوند و اگر عکسی برای نمایش دادن یافت نشد مقدار null در picturebox قرار می گیرد.

کد دکمه ی برو تو سبد:

```
if (e.ColumnIndex == ۱)

if (Int۳۲.Parse(textBox_Count.Text) < ۱)
{
label_Error.Show();
return;
}
if (Int۳۲.Parse(textBox_Count.Text) >
Int۳۲.Parse(dataGridView_TypeWriteList.Rows[e.RowIndex].Cells["Column_Count"].Value.ToString()))
{

", MessageBoxButtons.OK, فروشگاه", "تعداد وارد شده بیشتر از تعداد موجود است"MessageBox.Show(".
MessageBoxIcon.Exclamation);
textBox_Count.SelectAll();
textBox_Count.Focus();
return;
}
```

در این قسمت در ابتدا بررسی شده است که عکس با شماره ی موجود وجود داشته باشد و شماره ی عکس برابر با یک رشته ی تهی نباشد.

در این هنگام بررسی می شود که تعداد وارد شده بیشتر از تعداد موجود نباشد که در این حالت پیغام می دهد که تعداد وارد شده بیشتر از تعداد موجود است.

کد تابع update:

```
int Count = ٠;
Command.CommandText = "SELECT _COUNT FROM BASCKET WHERE BASCKETID='" + BascketID
+ "' AND ID='٠٤" + comboBox_Group.SelectedValue.ToString() +
dataGridView_TypeWriteList.Rows[e.RowIndex].Cells["Column_ID"].Value.ToString() + "'";
Command.Connection = Connection;
Connection.Open();
Reader = Command.ExecuteReader();
if (Reader.Read()) //---حالتی که رکورد قبلا درج شده و باید تعداد به روز رسانی شود
{
//-----افزودن تعداد به تعداد قبلی و به روز رسانی رکورد
Count = Int٣٢.Parse(Reader[٠].ToString()) + Int٣٢.Parse(textBox_Count.Text);
ira.Update_Record("BASCKET", "BASCKETID='" + BascketID + "' AND ID='٠٤" +
comboBox_Group.SelectedValue.ToString() +
dataGridView_TypeWriteList.Rows[e.RowIndex].Cells["Column_ID"].Value.ToString() + "'", "_COUNT",
Count.ToString());
}
```

کد تابع درج رکورد:

```
else //---حالتی که کالا قبلاً درج نشده است
ira.Insert_Record("BASCKET", "BASCKETID, ID, _COUNT, NAME, PRICE, Sood, SHOPID, GID,
MID", BascketID, "٠٤" + comboBox_Group.SelectedValue.ToString()+
dataGridView_TypeWriteList.Rows[e.RowIndex].Cells["Column_ID"].Value.ToString(),
textBox_Count.Text,
dataGridView_TypeWriteList.Rows[e.RowIndex].Cells["Column_NAME"].Value.ToString(),
dataGridView_TypeWriteList.Rows[e.RowIndex].Cells["Column_PRICE"].Value.ToString(), Sood, "٠٤",
comboBox_Group.SelectedValue.ToString(),dataGridView_TypeWriteList.Rows[e.RowIndex].Cells["Colum
n_ID"].Value.ToString());
Connection.Close();
//----نمایش پیام درج کالا در سبد
if (checkBox_GotoBascket.Checked)
{
string msg = "تعداد + textBox_Count.Text + "
dataGridView_TypeWriteList.Rows[e.RowIndex].Cells["Column_NAME"].Value.ToString() + "در سبد قرار
داده شد.";
MessageBox.Show(msg, "مدیریت", MessageBoxButtons.OK, MessageBoxIcon.Information,
MessageBoxDefaultButton.Button١, MessageBoxOptions.RtlReading);
}
}
}
}
```

این قسمت برنامه در ابتدا کالای مورد نظر را با اطلاعاتش نشان میدهد .اگر این کالا قبلا در سبد خرید درج شده باشد و اکنون تعداد دیگری به آن اضافه شده باشد در آن هنگام تعداد با تعداد قبلی جمع شده و به روز رسانی می شود.

اگر قبلا درج نشده باشد این تعداد در سبد خرید قرار می گیرد و در انتها با یک پیغام نشان می دهد که چه تعداد از کالای مورد نظر در سبد قرار داده شده است.

کد مربوط به وارد کرد تعداد کالای مورد نیاز:

```
private void textBox_Count_TextChanged(object sender, EventArgs e)
{
label_Error.Hide();
foreach (char chr in (sender as TextBox).Text)
if (chr < '۰' || chr > '۹')
{
(sender as TextBox).Text = "۰";
break;
}
if ((sender as TextBox).Text == string.Empty) (sender as TextBox).Text = "۰";
}

private void textBox_Count_KeyPress(object sender, KeyPressEventArgs e)
{
if ((e.KeyChar < '۰' || e.KeyChar > '۹') && e.KeyChar != '\b')
e.KeyChar = '\۰';
}
```

در این قسمت texbox مربوط به تعداد محدود به گرفتن عدد و backspace می باشد تا کاربر بتواند اطلاعات غلط خود را پاک کند.

همچنین کاربر نمی تواند از عملیات کپی برای گنجاندن رشته در این قسمت استفاده کند و اگر کاربر اطلاعات درست در این قسمت وارد نکند texbox مربوطه مقدار پیشفرض صفر به آن تبدیل می شود.

کد مربوط به راهنما:

```
private void linkLabel_Help_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
Form_BuyHelp Help = new Form_BuyHelp();
Help.ShowDialog();
}
}
}
```

کد مربوط به قسمت sendbacsket:

کد کسر تعداد فروخته شده از تعداد موجود:

```
private void UpdateCount(string _ShopID, string _GID, string _ID, string _Count)
{
//--- این تابع برای کسر تعداد فروخته شده از تعداد موجود تعریف شده است.
string Tablename = "";
switch (_ShopID)
{
case "01": Tablename = "BOOK"; break;
case "02": Tablename = "LEARNINGMEDIA"; break;
case "03": Tablename = "POOSTER"; break;
case "04": Tablename = "TYPEWRITE"; break;
}
SqlConnection Connection = new SqlConnection("Data Source=localhost;Initial Catalog=Shop;Integrated
Security=True");
SqlCommand Command = new SqlCommand();
SqlDataReader DReader;
Int64  I = 0;
Command.Connection = Connection;
Command.CommandText = "SELECT _COUNT FROM " + Tablename + " WHERE GID='" + _GID + "'
AND ID='" + _ID + "'";
Connection.Open();
DReader = Command.ExecuteReader();
if (DReader.Read())
{
I = Int64.Parse(DReader[0].ToString());
}
Connection.Close();
I -= Int64.Parse(_Count);
ira.Update_Record(Tablename, "GID='" + _GID + "' AND ID='" + _ID + "'", "_COUNT", I.ToString());
}
```

در ابتدا توسط یک دستور swich بررسی شده که خرید از کدام غرفه انجام شده است.بعد از آنکه شماره ی غرفه را مشخص کردیم یک sqlconnection برای ارتباط با پایگاه داده تعریف می کنیم و یک command برای اجرای دستورات.

سپس توسط دستور datareader تعداد رکورد هایی را می خوانیم که نام جدول و نام گروه آن با رکورد موردنظر هم خوانی داشته باشد.اگر چنین رکوردی پیدا شد تعداد آن از تعدادی که خواسته شده کم می شود و در آخر رکورد را update کرده و connection را می بندیم.

کد مربوط به ارسال سفارش برای کسانی که مشترک نیستند:

```
private void radioButton_NoUser_CheckedChanged(object sender, EventArgs e)
{
label_ID.Enabled = false;
label_Password.Enabled = false;
textBox_ID.Enabled = false;
textBox_Password.Enabled = false;
label_Name.Enabled = true;
textBox_Name.Enabled = true;
textBox_Name.Focus();
}
```

در این قسمت نام کاربری و شماره ی کاربری که برای کسانی که مشترک هستند غیر فعال شده و texbox مربوط به نام و نام خانوادگی فعال می شود.

کد مربوط به ارسال سفارش برای کسانی که مشترک هستند:

```
private void radioButton_IsUser_CheckedChanged(object sender, EventArgs e)
{
label_ID.Enabled = true;
label_Password.Enabled = true;
textBox_ID.Enabled = true;
textBox_Password.Enabled = true;
label_Name.Enabled = false;
textBox_Name.Enabled = false;
textBox_ID.Focus();

}
```

در این قسمت Texbox مربوط به نام و شماره ی کاربری فعال و texbox مربوط به نام و نـام خـانوادگی غیر فعال می شود.

تابع مربوط به فرم sendbascket:

```
private void Form_SendBascket_Load(object sender, EventArgs e)
{
Text += "                    شما سبد شماره:" + BascketID;
}
```

تابع مربوط به دکمه ی برو تو سبد:

```
private void button_Send_Click(object sender, EventArgs e)
{
label_EmptyBascket.Hide();
ira.Select_Record("BASCKET", "BASCKETID", out DS, "BASCKETID", "=", "AND", BascketID);
if (DS.Tables[0].Rows.Count < 2)
{
label_EmptyBascket.Show();
return;
}

if (radioButton_NoUser.Checked)
{
if (textBox_Name.Text == string.Empty)
{
MessageBox.Show("کنید وارد را خانوادگی نام و نام.", "فروشگاه", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
textBox_Name.Focus();
return;
}
```

در این قسمت برنامه در ابتدا پیغام خطا را مخفی کرده و سپس در یک دستور Sql به دنبال سبدی می گردیم که شماره ی آن با شماره ی مورد نظر یکی باشد.

سپس چک شده است که آیا خریدار مورد نظر مشترک فروشگاه هست یا خیر.اگر مشترک فروشگاه نباشد بررسی می شود که نام و نام خانوادگی را وارد کرده باشد.

```
Customer = textBox_Name.Text.Trim();
}
else
{
ira.Select_Record("ACCOUNT", "NAME, TAKHFIF", out DS, "TYPEINDEX, ID, PASSWORD", "=",
"AND", "03", textBox_ID.Text, textBox_Password.Text);
if (DS.Tables[0].Rows.Count < 1)
{
MessageBox.Show("است اشتباه رمز شناسه یا اشتراک شناسه.", "فروشگاه", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
textBox_ID.Focus();
return;
```

```
}
```

اگر خریدار مشترک فروشگاه باشد در این صورت در جدول account به دنبال نـام و درصـد تخفیـف و سایر مشخصات وی می باشد.اگر مشترکی با چنین مشخصاتی وجود نداشت در این صورت یک پیغام ظاهر می شود که نشانگر آن است که نام و شماره ی کاربری را اشتباه وارد کردید.

```
Takhfif = DS.Tables[0].Rows[0]["TAKHFIF"].ToString();
Customer = DS.Tables[0].Rows[0]["NAME"].ToString();
}
string Row = ira.showdate(DateTime.Today, true);
Row = Row[2].ToString() + Row[3].ToString() + Row[5].ToString() + Row[6].ToString() + BascketID;
SqlConnection Connection = new SqlConnection("Data Source=localhost;Initial Catalog=Shop;Integrated
Security=True");
SqlCommand Command = new SqlCommand();
SqlDataReader DReader;
int I = 0;
Command.Connection = Connection;
Command.CommandText = "SELECT ID FROM CUSTOMS WHERE ROW='" + Row + "'";
Connection.Open();
DReader = Command.ExecuteReader();
if (DReader.Read())
{
DReader.Close();
Command.CommandText = "SELECT MAX(ID) FROM CUSTOMS WHERE ROW='" + Row + "'";
DReader = Command.ExecuteReader();
if (DReader.Read())
I = Int32.Parse(DReader[0].ToString());
}
```

در این قسمت برنامه شماره ی سبد مشتری را تعیین می کنیم به این صورت کـه در ابتـدا زمـان فعلـی را برگردانده و کاراکتر های دوم ،سوم ،پنجم و ششم آن را ذخیره می کنیم.تا اینجای برنامه سال و مـاه آن را ذخیره کرده و در برای شماره ی سبد در نظر می گیریم.سپس در دستور بعـد شـماره ی سـبد را ذیـره کرده و در دستور بعد تعداد دفعات ساخته شدن سبد با همین شماره سبد را بررسی می کنیم.
سپس اطلاعات موجود را با هم در قالب یک رشته جمع کرده و به عنوان شـماره ی ارسـال بـه مشـتری نمایش می دهیم.

```
Connection.Close();
I++;
string ClassID = Row + I.ToString();
ira.Delete_Record("BASCKET", "BASCKETID, ID", "=", "AND", BascketID, "###");
ira.Select_Record("BASCKET", "NAME, _COUNT, PRICE, SOOD, SHOPID, GID, MID", out DS,
"BASCKETID", "=", BascketID);
if (DS.Tables[0].Rows.Count > 0)
{
for (int i = 0; i <= DS.Tables[0].Rows.Count - 1; i++)
{
//---- انبار در کالا تعداد رسانی روز به
UpdateCount(DS.Tables[0].Rows[i]["SHOPID"].ToString(), DS.Tables[0].Rows[i]["GID"].ToString(),
DS.Tables[0].Rows[i]["MID"].ToString(), DS.Tables[0].Rows[i]["_COUNT"].ToString());
```

```
//--- سفارشات جدول در سفارش درج
ira.Insert_Record("CUSTOMS", "CLASSID, ID, ROW, _DATE, CUSTOMER, NAME, _COUNT, PRICE, SOOD, TAKHFIF", ClassID, I.ToString(), Row, ira.showdate(DateTime.Today, true), Customer, DS.Tables[0].Rows[i]["NAME"].ToString(), DS.Tables[0].Rows[i]["_COUNT"].ToString(), DS.Tables[0].Rows[i]["PRICE"].ToString(), DS.Tables[0].Rows[i]["SOOD"].ToString(), Takhfif);
}
}
Form_ClassIDMsg ClassIDMsg = new Form_ClassIDMsg(ClassID);
ClassIDMsg.ShowDialog();
ira.Delete_Record("BASCKET", "BASCKETID", "=", "AND", BascketID);
this.Close();
}
```

بعد از اینکه شماره ی ارسال را به عنوان یک رشته نمایش دادیم .بعد از ارسال این رکوردی را که بـرای این مشتری تشکیل شده بود پاک می کنیم و در انتها تعداد کالا ها را مجددا به روز رسانی می کنیم.

کد مربوط به برنامه ی اصلی در فرم خریدار:

کد مربوط به فراخوانی نابع ira:

```
namespace Custom
{
public partial class Form_Main : Form
{
string ShopID = "01";
string BascketID = "";
IRA ira = new IRA();
DataSet DS = new DataSet();
//-----------------

public Form_Main()
{
InitializeComponent();
}
```

تا این قسمت برنامه یک متغیر رشته ایی به نام shopid که معرف شمارهی غرفه می باشد تعریف کرده و مقدار پیش فرض آن را غرفه ی شماره ی یک قرار داده ایم.

سپس با دستور بعدی تابع ira را فراخوانی کرده و یک ds برای نگهداری مقادیری خروجی تعیین کردیم.

```
private void radioButton_Book_CheckedChanged(object sender, EventArgs e)
{
groupBox_Shop.Text = "غرفه در خواهم می " + (sender as RadioButton).Text + " .کنم خرید";
ShopID = (sender as RadioButton).Tag.ToString();
}
```

سپس در فرم اولیه یک یک groupbox تعریف کرده و متن بالای آن را جمله ی "می خواهم در غرفه ی .... خرید کنم" گذاشتیم. که متن جای خالی حاوی غرفه ایست که کاربر با انتخاب آن قصد خرید از آن را دارد.

سپس متغیر shopid برابر با شماره ی غرفه ایی می شود که کاربر قصد خرید از آن را دارد.

```
private void linkLabel_Open_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
switch (ShopID)
{
case "01": Form_Book Book = new Form_Book(BascketID); Book.ShowDialog(); return;
case "02": Form_LearningMedia LearningMedia = new Form_LearningMedia(BascketID);
LearningMedia.ShowDialog(); return;
case "03": Form_Pooster Pooster = new Form_Pooster(BascketID); Pooster.ShowDialog(); return;
case "04": Form_TypeWrite TypeWrite = new Form_TypeWrite(BascketID); TypeWrite.ShowDialog();
return;
}

}
```

سپس توسط یک دستور swich مشخص کردیم که اگر غرفه ی ۰۱ انتخاب شد فرم مربوط به غرفه ی کتاب باز شود.اگر شماره ی ۰۲ انتخاب شد فرم مربوط به رسانه ی آموزشی باز شود.اگر شماره ی ۰۳ انتخاب شد فرم مربوط به پوستر و اگر ۰٤ انتخاب شد فرم مربوط به لوازم التحریر باز شود.

کد مربوط به login سیستم:

```
private void linkLabel_Login_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
SqlConnection Connection = new SqlConnection("Data Source=localhost;Initial Catalog=Shop;Integrated
Security=True");
SqlCommand Command = new SqlCommand();
SqlDataReader DReader;
int I = 0;
Command.Connection = Connection;
Command.CommandText = "SELECT BASCKETID FROM BASCKET";
Connection.Open();
DReader = Command.ExecuteReader();
if (DReader.Read())
{
DReader.Close();
Command.CommandText = "SELECT MAX(BASCKETID) FROM BASCKET";
DReader = Command.ExecuteReader();
if (DReader.Read())
I = Int32.Parse(DReader[0].ToString());
}
Connection.Close();
I++;
BascketID = I.ToString();
ira.Insert_Record("BASCKET", "BASCKETID, ID, NAME, PRICE, _COUNT, SOOD, SHOPID, GID,
MID", BascketID, "###", "###", "###", "###", "###", "###", "###", "###");
groupBox_Shop.Show();
groupBox_Login.Hide();
}
```

در ابتدای برنامه یک connection برای برقراری ارتباط با بانک اطلاعای تعریف کردیم.سپس یک command برای اجرای دستورات مشخص شده و در انتها از dreader برای خواندن رکوردها استفاده می کنیم.

برای دادن شماره ی سبد یک شماره از جدول مربوط به سبد انتخاب می کنیم.سپس Connection را باز می کنیم و شروع به خواندن رکورد ها می کنیم.

اگر رکوردی برای خواندن پیدا شد آنگاه dreader را می بندیم.

در خاصیت Commandtext از دستور command در قالب یک دستور sql بزرگترین شماره ی سبد را که تا کنون وجود دارد را انخاب می کنیم.سپس آن رشته را به عدد تبدیل کرده و یک شماره به آن اضافه می کنیم.به ایت ترتیب شماره ی سبد مشتری فعلی تعیین می شود.

کد مربوط به خروج از سیستم:

```
private void linkLabel_Out_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
ira.Delete_Record("BASCKET", "BASCKETID", "=", "AND", BascketID);
groupBox_Shop.Hide();
groupBox_Login.Show();
}
```

در این قسمت برنامه وقتی که کاربر دکمه ی خروج از سیستم را زد.شماره سبدی که به وی داده شده بود از او گرفته می شود و فرم مربوط به ورود به سیستم مجددا نمایش داده می شود.

کد مربوط به تابع خروج از برنامه:

```
private void Form_Main_FormClosing(object sender, FormClosingEventArgs e)
{
e.Cancel = true;
}
```

کد مربوط به زمان برنامه:

```
private void timer1_Tick(object sender, EventArgs e)
{
string date = ira.showdate(DateTime.Today, false);
string time = "ساعت: " + DateTime.Now.Hour.ToString() + ":" + DateTime.Now.Minute.ToString() + ":" +
DateTime.Now.Second.ToString();
toolStripStatusLabel_Now.Text = date + "  " + time;
}
```

کد مربوط به نمایش سبد:

```
private void linkLabel_ShowBascket_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
Form_Bascket Bascket = new Form_Bascket(BascketID);
Bascket.ShowDialog();
}
```

با انتخاب این گزینه سبد خرید شما نشان داده خواهد شد تا از خرید خود در فروشگاه آگاه شوید.

کد تابع مربوط به برو تو سبد:

```
private void linkLabel_Send_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
Form_SendBascket SendBascket = new Form_SendBascket(BascketID);
SendBascket.ShowDialog();
ira.Select_Record("BASCKET", "BASCKETID", out DS, "BASCKETID", "=", "AND", BascketID);
if (DS.Tables[0].Rows.Count < 1)
linkLabel_Out_LinkClicked(sender, e);
}
```

در این قسمت وقتی که گزینه ی برو تو سبد انخاب شد در ابتدا پیغام مربوط به آن نمایش داده می شود و سپس در جدول سبد به دنبال شماره ی سبد این مشتری می گردد.اگر چنین شماره ایی پیدا نشد از سیستم خارج می شود.

کد مربوط به نمایش راهنما:

```
private void linkLabel_Help_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
Form_BuyHelp Help = new Form_BuyHelp();
Help.ShowDialog();
}


}
}
```

کد مربوط به custom_srvice:

```
namespace Customs_Service
{
public partial class Form_Customs : Form
{
public Form_Customs()
{
InitializeComponent();
}

IRA ira = new IRA();
DataSet DS = new DataSet();
```

در ابتدا بر طبق روال تابع ira را فراخوانی کرده و سپس یک ds برای اعلام نتایج تعیین می کنیم.

کد مربوط به load شدن برنامه ی custom:

```
//-----------------------------------------------------
private void LoadCustom(string _ClassID)
{
if (_ClassID == string.Empty) return;
ira.Select_Record("CUSTOMS", "*", out DS, "CLASSID", "=", "", _ClassID);
if (DS.Tables[0].Rows.Count > 0)
{
label_Date.Text = "تاریخ:" + DS.Tables[0].Rows[0]["_DATE"].ToString();
textBox_Customer.Text = DS.Tables[0].Rows[0]["CUSTOMER"].ToString();
textBox_Takhfif.Text = DS.Tables[0].Rows[0]["TAKHFIF"].ToString();
Int64 Sum = 0;
Int64 Total = 0;
for (int i = 0; i <= DS.Tables[0].Rows.Count - 1; i++)
{
listBox_Row.Items.Add((i+1).ToString());
listBox_Name.Items.Add(DS.Tables[0].Rows[i]["NAME"].ToString());
listBox_Count.Items.Add(DS.Tables[0].Rows[i]["_COUNT"].ToString());
listBox_Price.Items.Add(DS.Tables[0].Rows[i]["PRICE"].ToString());
Sum = Int64.Parse(DS.Tables[0].Rows[i]["_COUNT"].ToString()) *
Int64.Parse(DS.Tables[0].Rows[i]["PRICE"].ToString());
Total += Sum;
listBox_Sum.Items.Add(Sum.ToString());
}
textBox_Total.Text = Total.ToString();
}
```

در تابع فوق ابتدا بررسی شده است که شماره ی سفارشی را که کاربر وارد کرده است صحیح می باشد یا خیر.

اگر شماره ی وارد شده توسط کاربر صحیح نباشد یه دستور وی ترتیب اثر داده نخواهد شد.

در غیر این صورت از جدول custom مشخصات مربوط به کاربر که حاوی نـام ،مقـدار تخفیـف و ... مـی باشد و به همین ترتیب مشخصات کالاهایی که سفارش داده به صورت نام ، تعداد و هزینه نمایش داده می شود و در انتها هزینه ی کل خرید وی با تخفیف و سود این خرید نمایش داده می شود.

نتیجه گیری

خوش بود گر محک تجربه آید به میان

علم و فن آوری بی تردید وقتی با تجربه و عمل همراه شود تاثیرات مثبت و حقیقی خود را نمایان می سازد . امروزه بحث آموزش بی کسب مهارت در حوزه های مختلف فن آوری بی معنی و ناقص و بی فایده به نظر می رسد .

من نیز از درس پروژه چیز های زیادی آموختم و به نظرم یکی از مهمترین درس هایی است که هر کس در دوران تحصیل خود سپری می کند .

در طول دوران نوشتن پروژه علاوه بر ASP.NET در زمینه C#.NET هم مهارت هایی را کسب کردم و احساس می کنم برای وارد شدن به بازار کار (در آینده) چشم اندازم وسیع تر شده است .

پیوست

منابع و مآخذ

۱.آموزش گام به گام C#.NET ،نویسنده:جان شارپ ، مترجمان:مهندس مهرداد توانا،مهندس عاطفه شیجونی. انتشارات:مؤسسه فرهنگی هنری نقش سیمرغ.

۲.راهنمای جامع C# ،نویسنده:Deitel & Deitel، مترجم:مهندس بهرام پاشایی،ناشر:جهان نو.

۳.sql server۲۰۰۰ ،نویسنده Rebecca m.riordan ،مترجم مانی قاسم نیا همدانی ،انتشارات ناقوس

دلم می خواست بهترین را به تو تقدیم کنم

اما تو....

در میان بهترین هایم بهترینی

و در میان بزرگترین هایم بزرگترین.

تقدیم به تو که دلم می خواست نان شادی هایم را با

تو قسمت کنم.